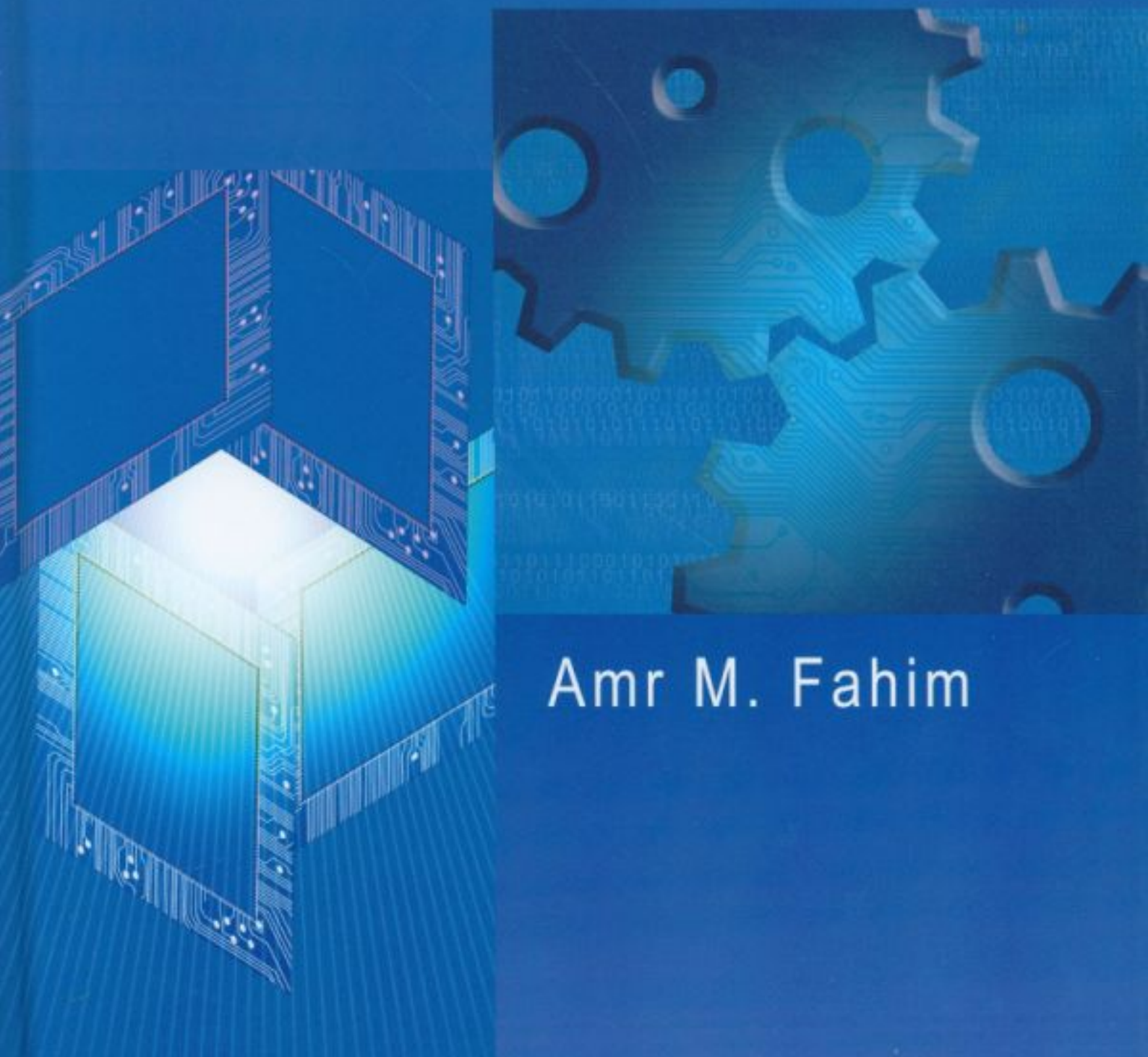
Clock Generators for SOC Processors

Circuits and Architectures



Amr M. Fahim

VISIT…

Clock Generators for SOC Processors

Amr M. Fahim

Clock Generators for SOC Processors

Circuits and Architectures

KLUWER ACADEMIC PUBLISHERS
Boston / Dordrecht / London

Library of Congress Cataloging-in-Publication Data

A C.I.P. Catalogue record for this book is available
from the Library of Congress.

ISBN-10 1-4020-8079-4 e-ISBN 1-4020-8080-8 Printed on acid-free paper.
ISBN-13 978-1-402-08079-1

© 2005 Kluwer Academic Publishers
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, Inc., 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now know or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks and similar terms, even if the are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1 SPIN 11117643

springeronline.com

Dedication

This book is dedicated to Sarah,
my parents, sister, and the
memory of my grandparents

Contents

ABOUT THE AUTHOR ..... **XIII**

PREFACE ..... **XV**

FOREWORD ..... **XVII**

1. INTRODUCTION **1**

1.1 WHAT ARE SYSTEM-ON-A-CHIP PROCESSORS? 1
1.2 ORGANIZATION 2

2. PHASE-LOCKED LOOP FUNDAMENTALS **3**

2.1 Introduction *3*
2.2 PLL Basics *3*
2.2.1 PLL Specifications 5
2.2.2 Charge Pump PLLs 6
2.2.3 PLL Building Blocks 7
2.3 Continuous-time Linear Analysis of PLLs *13*
2.4 Discrete-time Linear Analysis of PLLs *16*
2.5 Nonlinear Locking Behaviour of PLLs *18*
2.6 Summary *24*

3. LOW-VOLTAGE ANALOG CMOS DESIGN **27**

3.1 Introduction *27*
3.2 MOS Transistors *27*
3.3 Low-Voltage Current Mirrors *33*

3.4 Low-Voltage Charge Pumps *38*
3.5 Low-Voltage Oscillator Design *40*
3.5.1 Digital Tuning Circuitry 41
3.5.2 Ring Oscillator Design 43
3.5.3 LC VCO Design 51
3.6 Voltage and Current References *52*
3.7 Summary *58*

4. JITTER ANALYSIS IN PHASE-LOCKED LOOPS 61

4.1 Introduction *61*
4.2 Jitter Basics *61*
4.2.1 Definition of Jitter 61
4.2.2 Sources of Jitter 64
4.2.3 Jitter and Phase Noise 65
4.3 Jitter in Voltage Controlled Oscillators *67*
4.3.1 Classical Phase Noise Analysis 68
4.3.2 Cyclostationary Phase Noise Analysis 72
4.3.3 CMOS Voltage Controlled Ring Oscillators 75
4.3.4 Jitter in Differential LC VCOs 79
4.4 Jitter Performance of Closed-Loop PLL System *82*
4.4.1 Basic Linear Analysis 82
4.4.2 Effect of Varying Loop Parameters 83
4.4.3 Effect of PFD Nonlinearities on Jitter 86
4.5 Coupling Noise Effects on Jitter *89*
4.5.1 Power Supply Coupling Noise 89
4.5.2 Substrate Coupling Noise 93
4.6 Summary *95*

5. LOW-JITTER PLL ARCHITECTURES 99

5.1 Introduction *99*
5.2 Differential PLL Architecture *99*
5.2.1 Conventional Topology 99
5.2.2 Low-Noise Differential PLL Topology 106
5.3 Supply Voltage Regulated PLL Architectures *107*
5.3.1 Basic Concept 107
5.3.2 Regulator Topologies 108
5.3.2.1 Linear Voltage Regulator 110
5.3.2.2 Source Follower Voltage Regulator 112
5.3.2.3 Hybrid Voltage Regulation Approach 120

5.3.3 Performance Comparison 122
5.4 Adaptive PLL Architectures 123
5.4.1 Two-Mode Bandwidth Architecture 123
5.4.2 Process Insensitive Design 124
5.4.3 Analog Loop Bandwidth Controller 126
5.5 Resistorless Loop Filter PLLs 128
5.5.1 Basic Concept 128
5.5.2 Sample-Reset Loop Filter 128
5.6 Delay-Locked Loop Frequency Multipliers 129
5.6.1 DLL Operating Principle 129
5.6.2 DLL as a Frequency Synthesizer 131
5.6.3 Jitter Analysis 133
5.6.4 Hybrid PLL/DLL Approach 134
5.7 Summary 136

6. DIGITAL PLL DESIGN 141

6.1 Introduction 141
6.2 Basic Topology 141
6.3 Z-domain Analysis 144
6.3.1 Linear Model of DPLL 144
6.3.2 Linear Performance Evaluation 144
6.4 Circuit Implementation Issues 147
6.4.1 DAC optimization 147
6.4.2 Reference current temperature stability 149
6.4.3 Experimental Results 151
6.5 Alternate Digital PLL for Clock Generation 153
6.5.1 All-Digital PLL for Digitally Controlled Oscillator 153
6.5.2 All-Digital FLL with Phase Reset 156
6.6 Summary 159

7. DSP CLOCK GENERATOR ARCHITECTURES 163

7.1 Introduction 163
7.1.1 Why important? 163
7.1.2 Typical DSP Clock Generation Specifications 164
7.2 Sampling Clock Requirements for Data Converters 164
7.2.1 First order Analysis 164
7.2.2 General Case – A Qualitative Approach 166
7.2.3 General Case – A Quantitative Approach 167

7.2.4 Specific Case – Sigma-Delta A/D Converters 169
7.3 Jitter in Frequency Dividers 171
7.3.1 Uses of Frequency Dividers in SoC Processors 171
7.3.2 Mechanism of Jitter in digital frequency dividers 172
7.3.3 Synchronous Divider Model 174
7.3.4 Asynchronous Divider Model 175
7.3.5 Results 176
7.4 Fractional-N PLLs as Clock Generators 178
7.4.1 Basic Toplogy 179
7.4.2 Sigma-Delta PLLs 180
7.4.2.1 Basic Operation 180
7.4.2.2 Sigma-Delta Dithering 184
7.5 Oversampled PLL Topologies 184
7.6 Direct Digital Synthesis with Analog Interpolation 185
7.6.1 MNA Concept and Architecture 186
7.6.2 Parameter Analysis of MNA Counter 190
7.6.3 Circuit Implementation 191
7.6.4 Results 192
7.7 Summary 194

8. DESIGN FOR TESTABILITY IN PLLS 197

8.1 Introduction 197
8.2 Verification of SoC PLLs 197
8.2.1 Overview 197
8.2.2 Required PLL Tests 198
8.3 Jitter Measurement Techniques 201
8.3.1 High-Bandwidth ATE Jitter Measurement Setup 202
8.3.2 ATE Jitter Test Equipment 204
8.4 Design for Testability and Self-Test in PLLs 209
8.4.1 Functional Test 210
8.4.2 Indirect Jitter Measurement Techniques 210
8.4.3 Precise On-Chip Jitter Measurement 213
8.5 Summary 215

9. CLOCK PARTITIONING AND SKEW CONTROL 219

9.1 Introduction 219
9.2 Clock Distribution Networks in SoCs 220
9.3 Performance Limitations in Clock Networks 221
9.4 Active Skew Management Strategies 225

9.4.1 Effect of Clock Skew, Jitter on Power Consumption .225
9.4.2 Ad-hoc Deskew Strategies 228
9.4.3 DLL-based Deskew Strategies 229
9.5 Multi-phase Clock Generator 230
9.5.1 Passive Method 230
9.5.2 Master-Slave DLL 230
9.5.3 Pulsewidth Control Loops 233
9.6 Low-Power Clock Distribution Strategies 234
9.6.1 Active Clock Gating 234
9.6.2 Low-Power Gigahertz Clock Strategies 234
9.7 Multi-clock Domain Interfacing 235
9.8 Summary 238

About the Author

Amr M. Fahim received his B.A.Sc, M.A.Sc, and Ph.D. degrees from the University of Waterloo in Computer Engineering in 1996 and Electrical Engineering in 1997, and 2000 respectively.

During his graduate studies, he has had research collaboration twice with industry. In May - August 1996, he worked with Rockwell Semiconductor Systems (currently Mindspeed) in Newport Beach, California designing low-power PLL-based on-chip clock generators for microprocessors. In June - December 1999, he worked with Texas Instruments designing low-power RF fractional-N frequency synthesizers.

Upon completion of his Ph.D. degree in 2000, he joined Qualcomm Inc., San Diego, CA, where he spent four years driving the development of mixed-signal IC cores for clocking generator systems for integrated analog/digital Mobile Station Modem (MSM) chipsets. He is currently with Skyworks Solutions Inc. in Irvine, CA, working on mixed-signal and radio frequency integrated circuit (RFIC) designs. His research interests include design and analysis of low-power integrated frequency synthesis techniques and mixed-signal/RF system-on-a-chip (SoC) processors. He has authored over 25 papers and has several U.S. and European patents to his credit. He has also been a reviewer for numerous papers sent for publication including the IEEE Journal of Solid-State Circuits, IEEE Transactions on Circuits and Systems II, Midwest Symposium on Circuits and Systems, and VLSI Symposium.

Preface

This book examines the issue of design of fully integrated frequency synthesizers suitable for system-on-a-chip (SOC) processors. This book takes a more global design perspective in jointly examining the design space at the circuit level as well as at the architectural level. The coverage of the book is comprehensive and includes summary chapters on circuit theory as well as feedback control theory relevant to the operation of phase locked loops (PLLs). On the circuit level, the discussion includes low-voltage analog design in deep submicron digital CMOS processes, effects of supply noise, substrate noise, as well device noise. On the architectural level, the discussion includes PLL analysis using continuous-time as well as discrete-time models, linear and nonlinear effects of PLL performance, and detailed analysis of locking behavior.

The material then develops into detailed circuit and architectural analysis of specific clock generation blocks. This includes circuits and architectures of PLLs with high power supply noise immunity and digital PLL architectures where the loop filter is digitized. Methods of generating low-spurious sampling clocks for discrete-time analog blocks are then examined. This includes sigma-delta fractional-N PLLs, Direct Digital Synthesis (DDS) techniques and non-conventional uses of PLLs. Design for test (DFT) issues as they arise in PLLs are then discussed. This includes methods of accurately measuring jitter and built-in-self-test (BIST) techniques for PLLs. Finally, clocking issues commonly associated to system-on-a-chip (SOC) designs, such as multiple clock domain interfacing and partitioning, and accurate clock phase generation techniques using delay-locked loops (DLLs)

are also addressed. The book provides numerous real world applications, as well as practical rules-of-thumb for modern designers to use at the system, architectural, as well as the circuit level. This book is well suited for practitioners as well as graduate level students who wish to learn more about time-domain analysis and design of frequency synthesis techniques.

Foreword

Current literature is filled with textbooks and research papers describing frequency synthesizers from a front-end wireless transceiver perspective. The emphasis has historically been on evaluating the frequency synthesizer's performance in the frequency domain, i.e. in terms of phase noise and spurious signals. As microprocessor frequency surge, the need to understand digital requirements for low-jitter and the design of low-jitter frequency synthesizers and clock generator becomes every increasingly important. This book is dedicated to the time-domain (i.e. jitter) design and analysis of frequency synthesizers and clock generators for microprocessor applications. Such explanations are often scattered through literature and, to my knowledge, has not recently been gathered into one comprehensive textbook.

This book also focuses on the CMOS IC implementation of such synthesizers. An entire chapter is dedicated to low-voltage mixed-signal integrated circuit design in deep submicron CMOS technologies. Subsequent chapters discuss the design and analysis of the most common frequency synthesizer, the phase-locked loop (PLL), as well as state-of-the-art innovative architectures suitable for system-on-a-chip (SOC) processors. Design for Testability (DFT) is also discussed in the context of frequency synthesizers in SOC processors. The book concludes by discussing some of the most common issues that arise in clock interfacing, clock distribution, and accurate delay generation through delay-locked loops (DLLs) as they apply to SOC processors. Such issues mainly arise from having to communicate data and clock signals across multiple clock and power

domains. The book provides numerous real world applications, as well as practical rules-of-thumb for modern designers to use at the system, architectural, as well as the circuit level.

Chapter 1

INTRODUCTION

1.1 What are System-on-a-Chip Processors?

System-on-a-chip (SoC) technology is the integration of all the necessary electronic circuits and devices for a specified system into a single integrated circuit (IC) package. This typically includes an embedded microprocessor or controller, dedicated logic, memory, and some analog circuitry, such as analog-to-digital (A/D) converters or digital-to-analog (D/A) converters. In some cases, such as Bluetooth processors, this may include an entire RF wireless transceiver packaged into a single chip.

System-on-a-chip processors have been enabled by rapid advances in microelectronics, specifically complementary metal oxide semiconductor (CMOS) technology. The computing power of microprocessors has been increasing exponentially following Moore's Law. Such advances have brought new design challenges and opportunities that have never been previously anticipated. These design challenges include reducing power consumption, enhancing packaging density and reliability, and development of design automation tools to handle billions of transistors.

Advances SoC technology has already profoundly changed our society. Today mobile communication is affordable and available in most parts of the world. Wireless internet access is becoming a reality. Soon one will be able to browse the internet at tens of megabit per second data rates from a handheld device anywhere in the world.

The future of SoC technology offers new and radical concepts. As nanosensors are being integrated in SoC devices, nanorobots are soon to be a reality. Nanorobots may have profound medical applications such as creating programmable antivirus devices. Other nanoscale SoC devices can

have integrated video processors attached to optic nerve endings to enable vision to the blind.

One dominant trends of SoC processors is increased computing power, which entails increased clock speeds. This naturally makes clock partitioning and generation in SoC processors a very important topic. This book is dedicated to a detailed discussion on the design and analysis of clock generation units used in SoC processors. This book also addresses the issues of clock partitioning in SoC processors.

1.2 Organization

This book is divided up into nine chapters including this introduction chapter. Chapter 2 discusses phase-locked loop (PLL) fundamentals. PLLs are the most common implementation of a clock generation unit for SoC processors. Chapter 3 discusses the design challenges of PLL circuit blocks implemented in deep sub-micrometer CMOS technologies. Specifically, low-voltage CMOS analog design is addressed. Once PLL fundamentals and CMOS circuit implementation of PLLs have been discussed, chapter 4 provides a detailed jitter analysis in PLLs. This includes jitter definitions, power supply coupling noise effects on PLLs, oscillator jitter analysis, as well as jitter performance of closed-loop PLL systems. Chapter 5 explores the reduction of jitter in PLLs at the architectural level. Such architectures include supply voltage regulated PLL topologies, adaptive PLL topologies, and delay-locked loop (DLL) based frequency multiplication topologies. The performance of the various clock generation architectures is analyzed in detail and is compared with one another. In chapter 6, an alternative paradigm to PLL design is considered, where the loop filter is completely digitized. In chapter 7, clock generators that are specific to digital signal processing (DSP) applications are given. Modern techniques for design for testability (DFT) for PLLs are given. This includes methods of characterization and verification of embedded PLLs, jitter measurement techniques, and PLL built-in-self-test (BIST) techniques. Finally, chapter 9 concludes the book by a discussion on more global issues such as clock partitioning and skew control on clock signals in large SoC processors.

Chapter 2

PHASE-LOCKED LOOP FUNDAMENTALS

2.1 Introduction

Phase-locked loops, being the most commonly used form of clock generation used in SoC processors. Phase-locked loops (PLLs) can accurately generate a desired frequency that is a multiple of a high precision crystal reference. The main frequency synthesizer parameters include frequency range, frequency resolution, switching speed, jitter and power consumption. Reasons for their popularity include ease of integration and high frequency accuracy. In this chapter, the basic operating principle of PLLs is described. This includes performance metrics and a more detailed description of each block of the PLL. Linear and non-linear, continous-time and discrete-time modeling of PLLs are also covered.

2.2 PLL Basics

The operating principle of the PLL is first introduced by defining the basic structures in the loop. The PLL consists of a voltage-controlled oscillator (VCO), a frequency divider, a phase detector (PD), a loop filter, and a divide-by-2 circuit as shown in Figure 2-1.

The VCO produces a frequency proportional to its input voltage. More specifically,

$$\omega_2(t) = \omega_0 + K_0 u_f(t) \tag{2.1}$$

where ω_0 is the free oscillation frequency (frequency at which the VCO oscillates at when no input voltage is applied) measured in rads s^{-1}, and K_0 is the VCO gain (in $s^{-1}V^{-1}$). The phase detector compares the input signal, R,

and the feedback signal, V, and produces an output which is proportional to the phase difference between the two signals. The operation of the phase detector may be expressed as

$$u_d(t) = K_d\theta_e \tag{2.2}$$

where K_d is the gain of the phase detector and θ_e is the difference in phase between R and V (measured in rads). The units of K_d is V rad^{-1}. The purpose of the loop filter is two fold. First, it controls the bandwidth of the closed loop system. The loop bandwidth, in turn, controls the *settling time*, or *lock time* of the PLL. Secondly, it acts as to smoothen the output from the phase detector to provide a slow changing analog input to the VCO. The loop filter in most cases is a first or second order RC passive low pass filter. The frequency divider allows the input frequency to be lower than the output frequency by a factor of N. This means that, although the frequency of a crystal oscillator may be low (<40MHz), a PLL can be used to indirectly generate frequencies as high as several hundred megahertz. If the value of N is variable, then the PLL can generate programmable output frequencies.

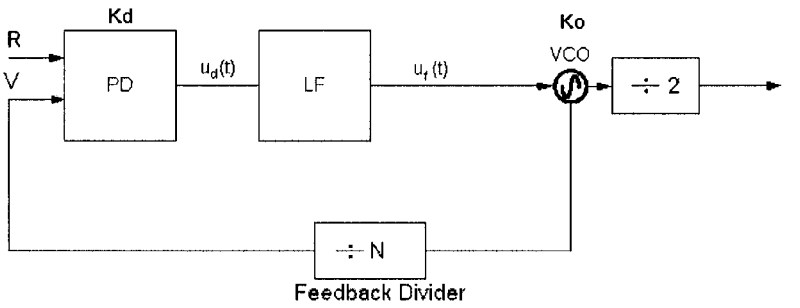


Figure 2-1. Basic PLL Configuration

The operation of the PLL is actually quite simple. If the input signal's phase is lagging behind the phase produced by the VCO, then this means that the VCO frequency is too high and hence must be lowered. The phase detector handles this by producing an inhibiting signal to the loop filter, which in turn lowers the voltage input to the VCO, $u_f(t)$. As can be shown by equation (2.1), lowering $u_f(t)$ amounts to lowering $\omega_2(t)$, the output frequency of the VCO, which is what is desired. On the other hand, if the input signal's phase is leading that of the VCO, the PD produces an excitatory signal to the loop filter which increases the output voltage of the loop filter, and in turn, increases the output frequency of the VCO, as desired. The divide-by-2 circuit at the output of the VCO is needed to provide a 50% duty cycle clock signal.

2.2.1 PLL Specifications

There are several figures of merit that objectively determine the PLL's performance. The three main performance metrics considered are *lock time*, *jitter*, and *power consumption*.

Lock time is the time it takes for the PLL to produce the desired output frequency to within a certain frequency error tolerance. Lock time is normally expressed as the amount of it takes for a PLL to produce an output frequency within a specified frequency accuracy. Frequency accuracy is expressed either in Hertz (Hz) or in parts per million (ppm). For example, if the required output frequency is 500MHz and the required frequency accuracy is 1kHz, then a lock time specification can read as 50μsec for 1kHz resolution, or 50μsec for 2ppm resolution. One of the most successful ways to reduce power in digital signal processors is to turn off either parts of the processor or the entire processor when not in use [1]. This may include turning off the PLL. Since the time it takes for the PLL to turn on again could take as much as tens or hundreds of microseconds, minimizing PLL lock time for SoC processors may be important.

One of the most important parameters in PLLs is jitter, or phase error. It is important to make a distinction between additive noise and phase error. A non-ideal sine wave may be given as

$$V(t) = V[1 + n_1(t)] \cdot \sin[\omega_t + n_2(t)] \tag{2.3}$$

Consider the sine wave shown in Figure 2-2(a). Due to thermal noise in the electronic devices in a PLL, the sine wave may be corrupted by both additive noise and phase error as shown in Figure 2-2(b) and 2.2(c), respectively. In the case of additive noise, only the amplitude is affected (i.e. $n_1(t)$ represents additive noise in the above equation); whereas in the case of phase error, horizontal excursions superimposed on the sine wave exist (i.e. this is represented by $n_2(t)$ in the above equation). This results in an error term associated with each period of the synthesized sine wave. For example, if a period of 5ns is required, at a certain instance a period of 5.1ns or 4.8ns may be produced; however, the average period produced by the PLL remains at 5ns. Jitter in PLLs will be analyzed in more detail in chapter 4.

One important parameter in PLLs is power consumption. This is especially important in portable applications, such as cellular phones and laptops equipped with wireless modems. Minimizing this parameter usually comes at the expense of increased jitter, as is demonstrated in chapter 4. Although the power consumption of the PLL is small compared to modern microprocessors, it may still be important to minimize. For SoC processors, for example, it may be necessary to keep the PLL on to provide clock signal

for DRAM refreshing or immediate interrupt handling. This means that the PLL may be part of the processor's standby power budget. Also, large SoCs may mandate the use of several PLLs on-chip used simultaneously. Power minimization in PLLs for such applications is discussed in chapters 7 and 9 in more detail.

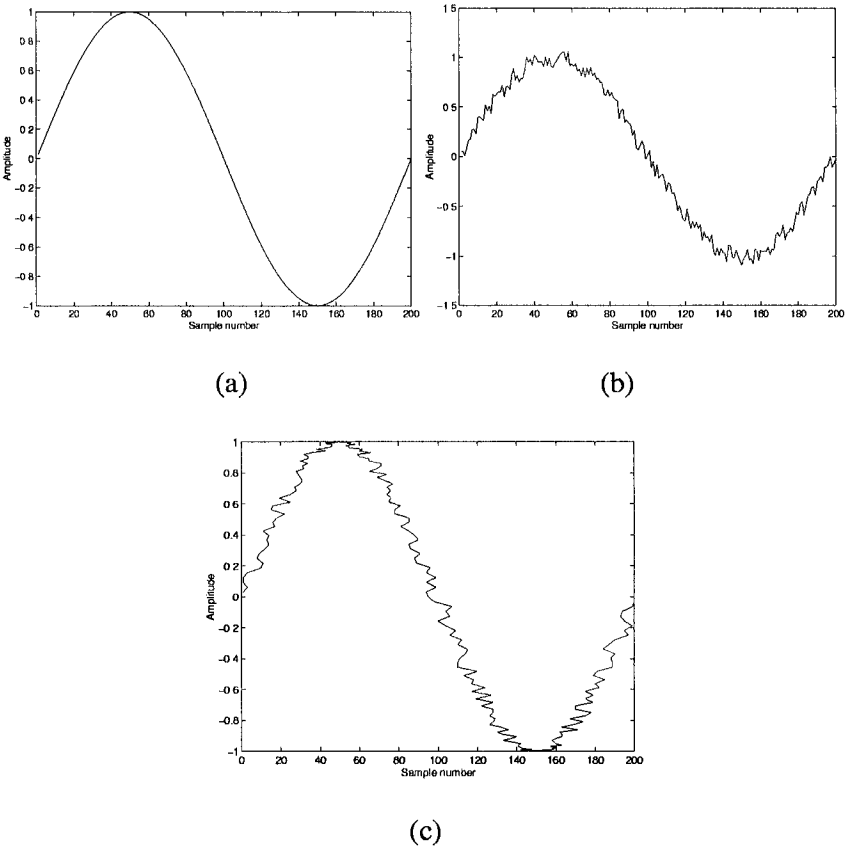


Figure 2-2. Sine wave in (a) ideal form, (b) corrupted by additive noise, (c) or corrupted by jitter.

2.2.2 Charge Pump PLLs

In order to have an infinite capture range, an active filter must be used for the loop filter. Unfortunately, this is not a suitable structure for low-jitter applications since an active loop filter introduces a significant amount of noise. Alternatively, a charge-pump may be employed [2]. Figure 2-3 shows a block level diagram of a charge-pump based PLL. The reason for its popularity includes extended lock range and low cost [2]. As its name

implies, a charge pump is responsible for injecting a constant amount of current to the loop filter for a time period proportional to the phase error. Its input is from the phase detector. If UP=1, then current is injected into the loop filter, and hence the voltage across the loop filter is increased. If DN=1, on the other hand, an equal amount of current is drawn out of the loop filter, and thereby decreasing the voltage across the loop filter. Unless otherwise stated, all PLLs from this point forward are assumed to charge-pump based PLLs.

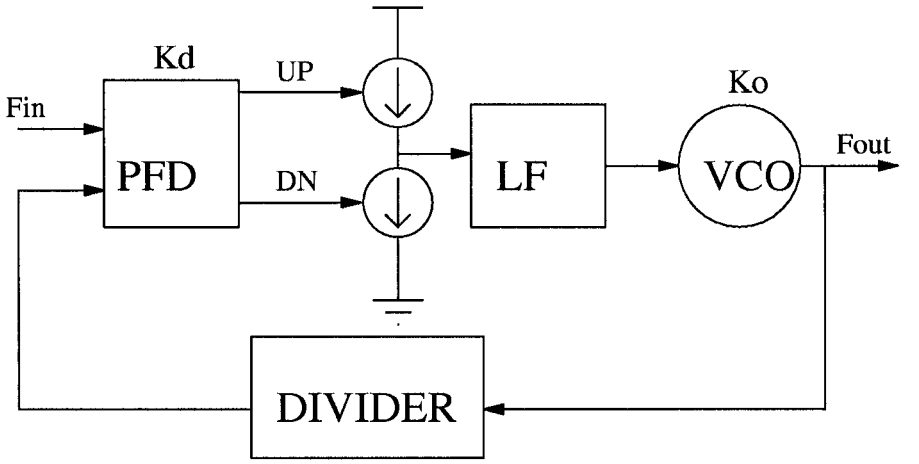


Figure 2-3. Charge pump based PLL

2.2.3 PLL Building Blocks

A practical implementation of a fully monolithic modern microprocessor PLL is shown in Figure 2-4. As the figure shows, the PLL consists of a phase-frequency detector (PFD), a charge pump, passive loop filter, voltage-to-current converter (V2I), a current controlled oscillator (ICO), a feedback frequency divider (FD), a divide-by-2 (div2) output divider, an on-chip current reference generator.

Phase-frequency Detector. A phase-frequency detector is a special type of a phase detector. A phase detector, as its name implies, is capable of detecting a phase difference between two inputs. The output is proportional to the phase difference. The simplest implementation of a phase detector is an XOR gate. Due to its symmetric operation, the XOR gate is not capable of detecting phase differences greater than 180 degrees, and hence frequency differences of more than one beat note apart cannot be detected.

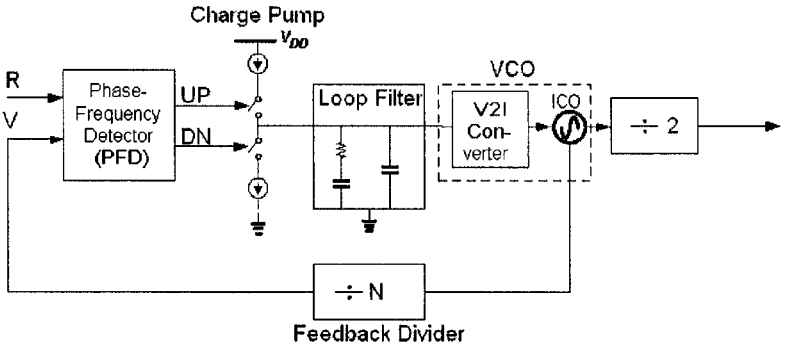


Figure 2-4. Practical fully monolithic microprocessor PLL

A phase-frequency detector (PFD), on the other hand, is capable of detecting both phase and frequency differences. A conceptual diagram of the PFD is shown in Figure 2-5. Input V is the signal from the frequency divider and R is the reference signal (external input). If R lags behind V, then this means that the signal generated by the VCO has too high of a frequency. In this case, the DN signal is enabled. The DN signal goes into a device called a *charge pump*, which lowers the voltage across the loop filter, which is also the input voltage of the VCO. Hence, the frequency of the VCO is lowered as desired. A similar explanation holds for the case where R leads V.

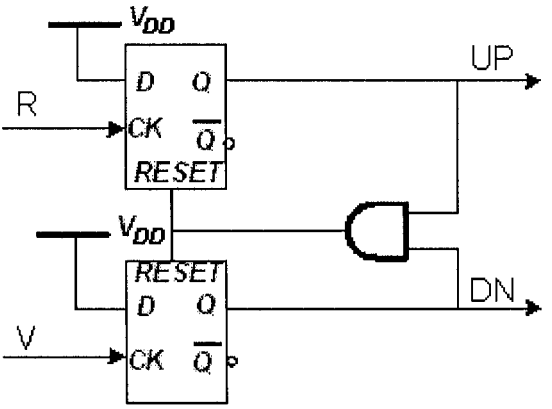


Figure 2-5. Conceptual diagram of a PFD

Note that since there are two flip-flops, there are a total of four possible states. One state, which is when both flip-flops are set, is eliminated by feeding the signals from both flip-flops' outputs to an AND gate which in

turn resets both flip-flops. The three valid states are listed in Table 2.1 below.

Table 2.1. Valid phase-frequency detector states

State #	State Condition	Meaning
-1	DN=1, UP=0	VCO freq too high
0	DN=0, UP=0	PLL is in phase lock
1	DN=0, UP=1	VCO freq too low

It is the fact that there are different states for when the VCO frequency is too high or low that gives the PFD memory of which state it was in. It is this memory capability which gives the PFD its ability to frequency lock. The linear operation of the three state PFD is 360 degrees.

One important issue concerning PFDs is the phase resolution that they are able to detect. For example, if the PFD can detect phase differences between $u_1(t)$ and $u_2(t)$ that are at least 100 picoseconds apart, then the gain of the PFD drops dramatically for phase errors of less than 100 picoseconds. The region in which the PFD cannot distinguish between the phases of the two input signals is known as the *deadzone band* [3]. In this region, the PLL no longer behaves as a linear system and noise performance of the PLL can be significantly degraded. Unfortunately, when the PLL is locked, the phase difference between the two inputs of the PFD is minimum. Figure 2-6 shows a typical transfer function of a PFD illustrating the effect of a deadzone band.

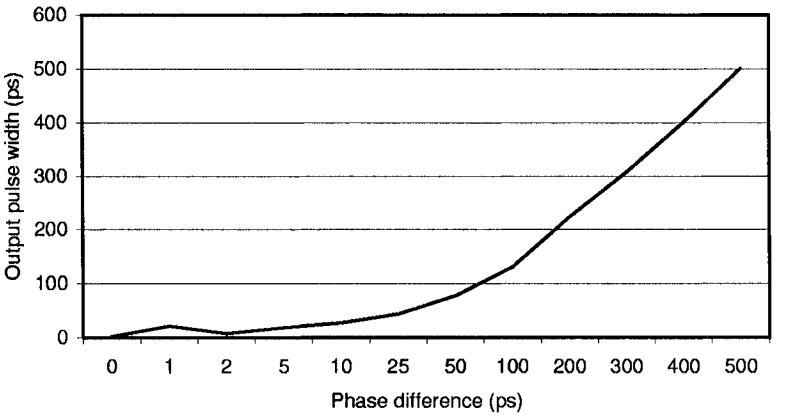


Figure 2-6. Transfer function of a PFD illustrating a deadzone band

One method of avoiding this problem altogether is by using a type-4 PFD [4], as shown in Figure 2-7. This is similar to a conventional PFD except that there is a delay on the feedback reset path. Assuming the PLL is in lock, the PFD will always send two short UP and DN pulses simultaneously at every positive edge of R and V. Assuming that a small differential phase develops between R and V, the fall times of the two pulses may be much less than one gate delay. Using this technique, arbitrarily small timing resolution can be achieved. Note that the deadzone band may still exist if slow circuit techniques are used, or if the circuit delay paths between the input and the UP and DN outputs are not balanced.

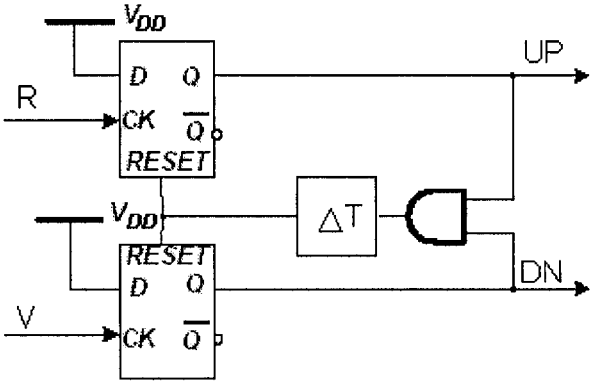


Figure 2-7. Type-4 PFD schematic diagram

To illustrate the operation of the type-4 phase detector, Figure 2-8(a)-(c) show the HSPICE simulation output for the three cases of the PLL being in lock, VCO being too slow, and too fast, respectively. Note that both UP and DN pulses are generated simultaneously even if the PLL is locked. Note the minimum finite pulse width of both UP and DN pulses in Figure 2-8(c). Optimization of this pulse width is discussed in more detail in Section 4.4.4.

Charge-Pump. Charge pumps are used quite often in frequency synthesizer applications. Reasons for their popularity include extended lock range and low cost [2]. As shown in Figure 2-3, a charge pump lies between the phase detector and the loop filter. As its name implies, a charge pump is responsible for injecting a constant amount of current to the loop filter. If UP=1 (in the PFD), then current is injected into the loop filter, and hence the voltage across the loop filter is increased. If DN=1, on the other hand, an equal amount of current is drawn out of the filter, and thereby decreasing the voltage across the loop filter. Circuit implementations of charge pumps are discussed in more detail in chapter 3.

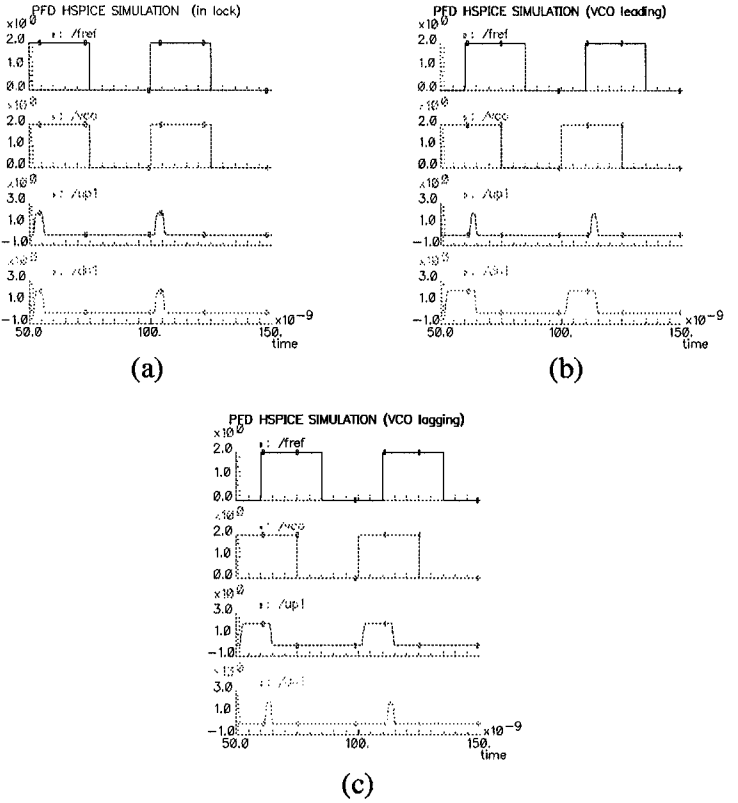


Figure 2-8. HSPICE waveforms of type-4 PFD when the PLL is (a) in lock, (b) lagging in frequency, (c) leading in frequency.

Loop Filter. The PLL's loop filter largely affects the PLL's dynamics, performance, and area. The loop filter is typically implemented as a passive loop filter. Current injected into the loop filter is converted into a voltage by means of a capacitor. This voltage is then used to control the voltage-controlled oscillator (VCO). A typical loop filter is shown in Figure 2-9. The resistor is required for stability since it produces a zero in the second order continuous-time linear closed loop model of the PLL, as is demonstrated in section 2.3. A secondary capacitor is used to filter out the ripples produced by periodic injection of charge by the charge pump. The main capacitor is usually fairly large and can occupy as much as one-half of the PLL's total area. Although active loop filters may be used to reduce the PLL's area, they are typically not employed due to their inferior noise performance. For SoC applications, the loop filter is fully integrated on-chip.

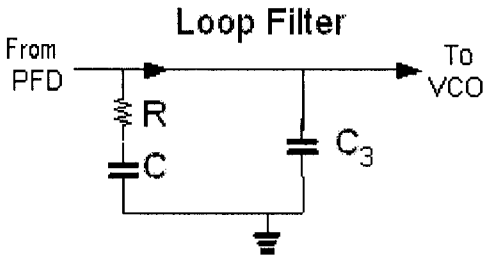


Figure 2-9. Typical loop filter for charge-pump based PLL

Voltage-controlled Oscillator. Monolithic PLLs for SoC applications usually employ a ring oscillator-based voltage-controlled oscillator (VCO). Furthermore, the VCO is broken into two stages, the voltage-to-current converter (V2I), and the current-controlled oscillator (ICO). This is done since the gain as well as the frequency of a ring oscillator is more easily controlled using a current control signal rather than a voltage control signal. A typical ring-oscillator based VCO is shown in Figure 2-10 [5].

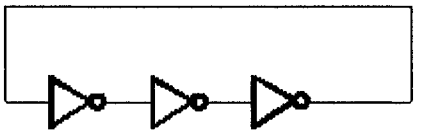


Figure 2-10. Ring-oscillator based VCO

The number of stages used in the ICO is of importance. For minimal power consumption, minimum number of stages is desired. For an ICO to oscillate, it must be capable of producing a 360° phase shift. If inverter stages are used, this means that each inverter must produce 180°/N, where N is the number of stages. For most CMOS technologies, 60° phase shift per stage is easily achieved, whereas 90° becomes more difficult. Also, as is shown in Section 4.3.3, minimal number of stages is desired for minimal jitter. For these two reasons, it is now widely accepted that the optimal number of stages in a ring oscillator is three.

Frequency dividers. Frequency multiplication is achieved in PLLs by dividing down the VCO signal with a fixed number, N, and comparing it with the PLL's input reference frequency. In SoC PLLs, this ratio is usually an integer. The most common method of dividing down the VCO frequency is through the use of an asynchronous counter [6], as shown in Figure 2-11. The asynchronous counter is first loaded with the desired frequency division value, N. The asynchronous counter consists of a chain of divide-by-2 circuits. The output of the first stage feeds the clock signal of the second stage. This configuration enables down counting. The logic AND block

detects a zero condition, which means that the count operation has completed. Since the output of one divide-by-2 stage feeds the clock input of the subsequent stage, the jitter is accumulated from one block to the next. In order to eliminate this accumulation of jitter, a retiming flip-flop clocked with the VCO signal is used. The output of the AND logic feeds the retiming flip-flop. On the negative edge of the clock, the frequency divider output toggles. The output signal is also used as a "load" signal in order to reinitialize the count operation.

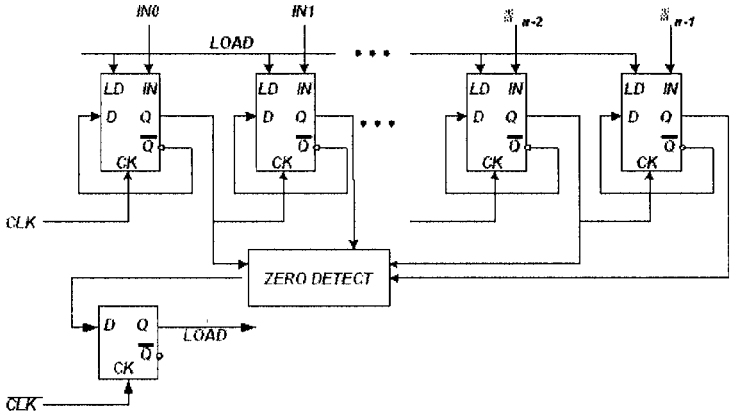


Figure 2-11. Asynchronous counter with retiming

2.3 Continuous-time Linear Analysis of PLLs

For the classical charge pump PLL, shown in Figure 2-3, it is instructive to obtain the various transfer functions of the PLL. Assuming that the closed loop bandwidth of the PLL is less than the reference frequency by at least an order of magnitude and the PLL is in near locked condition, the PLL may be thought of as a continuous-time linear system. The transfer functions are given as:

$$H_1(s) = \frac{K_d K_o F(s)}{s + K_d K_o F(s)/N} \quad \text{from input to VCO output} \quad (2.4)$$

$$H_2(s) = \frac{sK_d F(s)}{s + K_d K_o F(s)/N} \quad \text{from input to LF output} \quad (2.5)$$

$$H_3(s) = \frac{sK_d}{s + K_d K_o F(s)/N} \quad \text{from input to PFD output} \quad (2.6)$$

$$H_4(s) = \frac{K_o F(s)}{s + K_d K_o F(s)/N} \quad \text{from PFD to VCO output} \quad (2.7)$$

$$H_5(s) = \frac{K_o}{1 + K_d K_o F(s)/N} \quad \text{from LF to VCO output} \tag{2.8}$$

These equations may be used to evaluate the effect of injecting noise at different points in the PLL on the output response of the PLL.

Equation (2.4) is the basic PLL transfer function. Assuming a charge-pump PLL with a first order loop filter, $H_1(s)$ may be rewritten as:

$$H_1(s) = \frac{I_p K_o (R + \frac{1}{sC})/2\pi}{s + \frac{I_p}{2\pi} K_o (R + \frac{1}{sC})/N} \tag{2.9}$$

Equation (2.9) is a second order system. The natural frequency and damping factor are then given as

$$\omega_n = \sqrt{\frac{K_o I_p}{2\pi NC}} \tag{2.10}$$

$$\zeta = \frac{\omega_n RC}{2} \tag{2.11}$$

The 3-dB bandwidth of the 2nd order PLL is given as [7]

$$f_{3dB} = \frac{\omega_n}{2\pi}\sqrt{2\zeta^2 + 1 + \sqrt{(2\zeta^2 + 1)^2 + 1}} \quad \text{Hz} \tag{2.12}$$

Since the loop filter introduces one pole and the VCO acts as a second pole, both the frequency and phase are maintained at a constant value during locked conditions.

As long as the PLL is within the lock-in range, the PLL can lock to within one beat note. The lock-in range is given as

$$\Delta\omega_L = 4\pi\zeta\omega_n \tag{2.13}$$

Using equations (2.13) and (2.14), the lock-in range can also be expressed as

$$\Delta\omega_L = \frac{RK_o I_p}{N} \quad \text{Hz} \tag{2.14}$$

As long as the frequency deviation is less than the lock-in range, the PLL will lock within one beat note, which is given as

$$T_P = \frac{(\Delta\omega)^2}{2\zeta\omega_n^3} \tag{2.15}$$

where $\Delta\omega$ is the step frequency change in rads/sec. Using equations (2.13) and (2.14), the lock-in time can also be expressed as

$$T_P = \frac{(\Delta f)^2}{\frac{R}{C}\left(\frac{K_o I_p}{N}\right)^2} \text{seconds} \tag{2.16}$$

where Δf is the step frequency change in hertz.

A second order loop filter is usually employed, as shown in Figure 2-9. The transfer function of such as system is given as

$$H(s) = \frac{K\left(\frac{b-1}{b}\right)\left(s+\frac{1}{\tau}\right)}{\frac{s^3\tau}{b}+s^2+K\left(\frac{b-1}{b}\right)s+\frac{K(b-1)}{b\tau}} \tag{2.17}$$

where $b=1+C/C_3$ and $K=K_oK_d/N$, and $\tau=RC$. The extra capacitor, C_3, is usually sized to be ten times smaller than the main loop filter capacitor, C. This is done to move the extra pole introduced by capacitor C_3 far away from the closed loop bandwidth such that it does not significantly affect the loop dynamics. This helps to maintain system stability as well as to validate all the preceding second order closed loop analysis.

The main reason for using such a capacitor is to dampen the voltage spikes that may otherwise occur in the loop filter after a charge pump output pulse. For fast charge pump current transients, current flows through the resistor and the main capacitor is effectively a short circuit. This creates a voltage drop across the resistor equal to $I_p \cdot R$. When the charge pump pulse ends, the current across the resistor becomes zero, and the loop filter voltage jumps to the capacitor voltage. When an additional capacitor C_3 is added to the loop filter, this spike is dampened by the low pass RC_3 filter response.

The effect of adding capacitor C_3 on the PLL's stability is best analyzed through root locus plots. The open-loop transfer function for any PLL is

$$G(s) = \frac{KF(s)}{s} \tag{2.18}$$

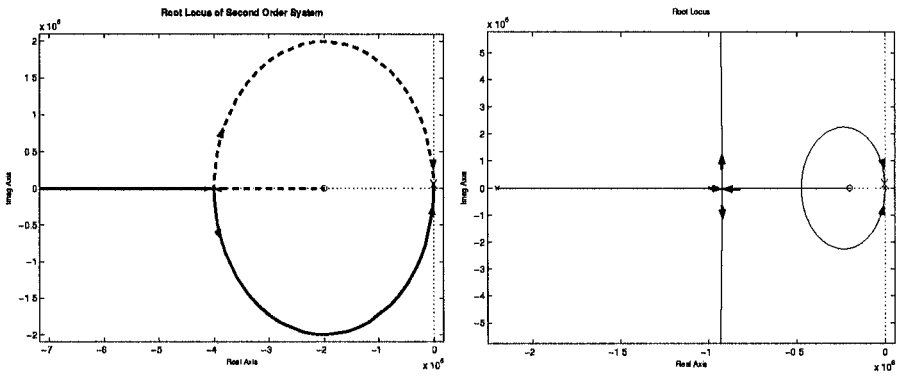
where $K= K_oK_d/N$ and F(s) if the loop filter's transfer function. The 1/s term introduces a pole at the origin and a zero at infinity. In the case of a second order system, the open loop transfer function is

$$G_2(s) = \frac{K(RCs+1)}{s^2} \tag{2.19}$$

This shows that the first order filter introduces a zero at 1/RC and another pole at the origin. In the case of a third order system, the open loop transfer function is

$$G_3(s) = \frac{K(RCs+1)}{s^2(RCC_3s+(C+C_3)} \tag{2.20}$$

This system shows two poles at the origin, one pole at $(C+C_3)/RCC_3$, and one zero at 1/RC. The root locus plots [8] of second order and third order systems are shown in Figure 12(a) and (b). The only difference between the two systems is the addition of the C_3 capacitor, which was ten times smaller than C. As the figures show, the third order system behaves the same as a second order system if the open loop gain (controlled by K) is small. For very large values of K, the system starts to become oscillator, although strictly speaking it is unconditionally stable.



(a) (b)

Figure 2-12. Root locus plots for (a) second order system and (b) third order system

2.4 Discrete-time Linear Analysis of PLLs

In the previous analysis, it was assumed that the PLL could be modeled as a continuous time linear system. Generally, this assumption is true for input and output sine waves [9]. For SoC processors, the input and output signals are square waves. The phase detector operates on the edges of the input square waves, which makes it a sampled system. For this reason, the PLL is essentially a discrete-time system. However, when the comparison frequency (sampling time of the discrete-time system) is relatively high with respect to the PLL's closed loop bandwidth, the PLL behaviour can be approximated as a continuous-time system. In the case of clock generators for SoCs, fast lock is required and this usually implies large loop bandwidth. When the closed loop bandwidth approaches the phase detector's comparison rate, the discrete-time behaviour of the PLL becomes important. In this section, the PLL is analyzed as a discrete-time linear system.

The main challenge of discrete-time analysis of PLLs is the accurate modeling of continuous time sub-systems such as the VCO and loop filter. If one considers the input to the loop filter and VCO to be a series of discrete

weighted current pulses from the charge pump, continuous-time to discrete-time transformation of these two sub-systems becomes relatively simple by using the impulse invariant transformation [10]. The impulse invariant transformation guarantees the accuracy of the discrete-time representation of a continuous-time waveform by sampling at a fixed time intervals. The impulse invariant response could be calculated by using the following steps:

1. Expand H(s) by method of partial fractions
2. Find h(t), the inverse Laplace Transform of H(s)
3. To guarantee accuracy at fixed time intervals of period T, let h[n]=h(nT)
4. Find the Z-transform of h[n]

Using the steps listed above, one can easily calculate the open loop and closed loop transfer functions of the PLL. Figure 2-13 below compares the continuous-time to the discrete-time frequency transfer functions of a second order PLL. As the figure reveals, the discrete-time response of the PLL shows a larger closed loop bandwidth than its continuous-time model. Figure 2-14 compares the predicted closed loop bandwidth of a continuous-time model of a PLL with its discrete-time counterpart. As the figure reveals, the discrepancy in predicted bandwidth grows with larger loop bandwidth.

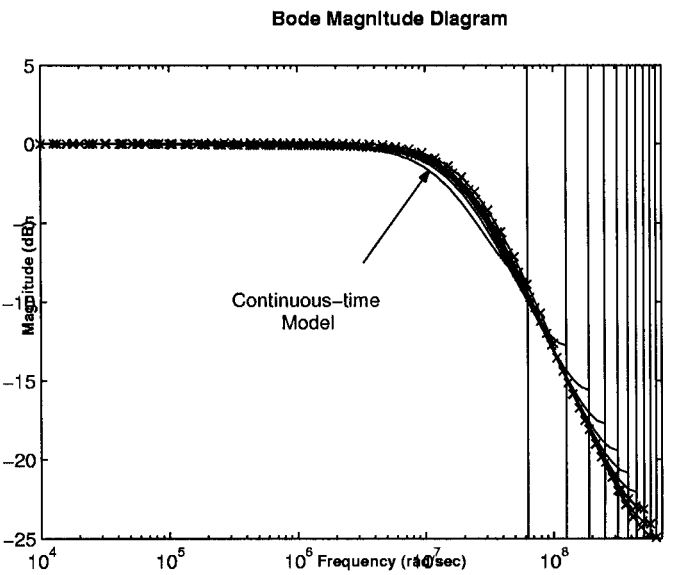


Figure 2-13. Frequency response of continuous-time versus discrete-time PLL models

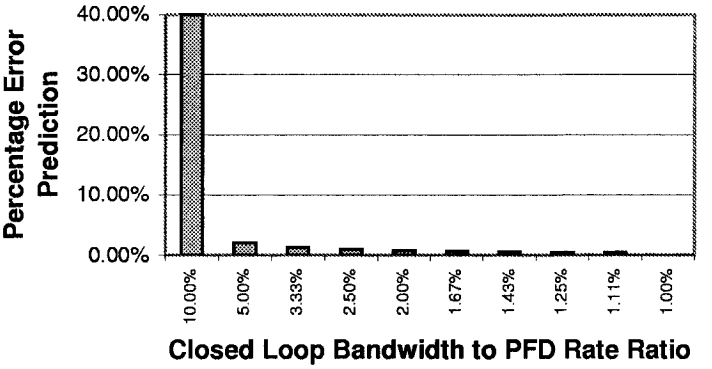


Figure 2-14. Prediction error in closed loop bandwidth of continuous-time linear model

2.5 Nonlinear Locking Behaviour of PLLs

The locking characteristics of a PLL can be divided into two categories: frequency locking and phase locking. Frequency locking is a highly nonlinear mechanism and depends strongly on the type of phase detector and loop filter used [11]. A phase-frequency detector (PFD) and a charge pump are assumed throughout this section. Figure 2-15 shows the typical locking behaviour of a charge pump based PLL. If the control voltage is much lower than its final value, this means that the period of the "V" signal is many times larger than the "R" signal. In this case, the UP signal may be asserted for more than one "R" period. When the UP signal is deasserted by the arrival of a "V" edge, the control voltage is held constant before it can be raised (or lowered) to another value.

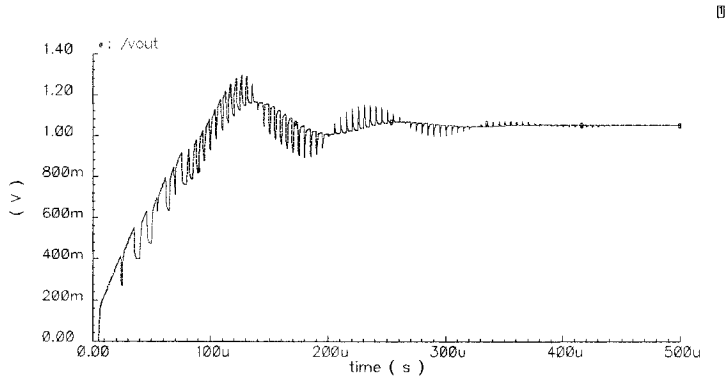
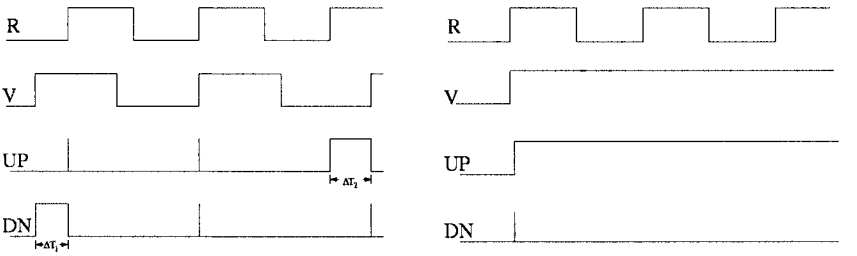


Figure 2-15. Typical locking behaviour of a charge pump based PLL

Two cases are shown in Figure 2-16. The first case, Figure 2-16(a), shows the simple case when the UP signal is monotonic and always asserted. Figure 2-13(b), on the other hand, shows the case where the "V" period is larger than the "R" period, but the DN signal is asserted for a certain amount of time. This is caused by a combination of two mechanisms. If the frequencies are different and the phases seem to line up, the UP signal pulses will diminish for a small time interval, then increase again once the PFD detects the differences in frequency. This phenomenon, which is referred to as "cycle slipping" [12]. During cycle slipping, the DN pulses are never activated. However, due to the finite delay in the PFD's feedback reset signal, both UP and DN can be high and hence, a "R" pulse can be missed during the reset pulse [14]. It is this mechanism that causes the DN signal to be activated and causes the rippling waveform in the control voltage during frequency locking, as shown in Figure 2-15.



(a) (b)

Figure 2-16. Two cases during frequency lock (a) in phase signals and (b) output phase signals

The transition from frequency to phase lock is marked when the control voltage first reaches its final value. The control voltage continues to oscillate around its final value several times due to the fact that the phases of the R and V signals were not initially matched. Recall that loop filter transfer function from PFD to the VCO output is given as

$$H_4(s) = \frac{K_o F(s)}{s + K_d K_o F(s)/N} \tag{2.21}$$

Assuming that the initial control voltage is denoted as V_o and the final value of the control voltage is denoted as V_f, the following relationship holds

$$V_f - V_o = I_p \Im^{-1}\{H_4(s)\}\Big|_0^{T_{FL1}} \tag{2.22}$$

where $\Im^{-1}\{\bullet\}$ indicates the inverse Laplace transform and T_{FL1} is the frequency lock time assuming the condition shown in Figure 2-16(a).

Assuming the more complicated condition shown in Figure 2-16(b), the charge pump initially drives the VCO in the wrong direction due to the fact that the V signal leads the R signal. As shown in Figure 3.13, the difference in the periods of the R and V signals is given as ΔT_1. After two R periods, the R signal leads the V signal and the UP signal drives the VCO in the correct direction. Note that since the last operation caused the VCO to slow down, the period of the V signal is now larger, and hence the UP pulse is larger than the previous down pulse. This causes the VCO to be steered in the correct direction after two R periods. In this case, the difference in VCO frequency over two R periods is given as

$$\Delta f\Big|_{2T_{REF}} = I_p\left(-\Im^{-1}\{H_4(s)\}\Big|_0^{\Delta T_1} + \Im^{-1}\{H_4(s)\}\Big|_0^{\Delta T_2}\right) \tag{2.23}$$

where

$$\Delta T_1 = T_V - T_R \tag{2.24}$$

and

$$\Delta T_2 = \frac{1}{\frac{1}{T_V} - I_p \Im^{-1}\{H_4(s)\}\Big|_0^{\Delta T_1}} - T_R \tag{2.25}$$

This pull-in process continues until the PLL reaches the lock-in range (defined by equation (2.13)). This means that the total change in frequency before lock-in range can be given as

$$\Delta f_{tot} = \sum_{i=1}^{n}(-1)^i I_p \Im^{-1}\{H_4(s)\}\Big|_0^{\Delta T_i} \tag{2.26}$$

where i is summed until the PLL is in lock-in range. Note that for each summation iteration i, a time period of $2T_{ref}$ elapses. Therefore, the frequency lock time can be given as

$$T_{FL2} = 2T_{ref}\, n \tag{2.27}$$

where n is the number of charge pump samples required to reach the lock-in range and T_{ref} is the period of the R signal. The total frequency lock in time is bounded by equations (2.22) and (2.27).

Phase locking is a more well understood phenomenon. During phase locking, the PLL transient response behaves as a linear system. Since the transfer function of the PLL linear model is known, its step response should give the correct phase locking characteristics. The phase lock time can be approximated as [13]

$$T_{PL} = \frac{2\pi}{\omega_n} \tag{3.23}$$

If a first order loop filter is considered, then the phase lock time can also be expressed as

$$T_{PL} = 2\pi\sqrt{\frac{2\pi NC}{K_o I_p}} = \frac{\pi RC}{\zeta} \tag{3.24}$$

where ω_n is the natural frequency and ζ is the damping factor.

An important issue is how much phase lock time is required if the control voltage is near its final value. In order to address this issue, several simulations were run in order to characterize the lock time behaviour of charge-pump based PLLs in more detail. In all these simulations the PLL parameters listed in Table 2.2 were used.

Table 2.2. PLL parameters used for phase lock characterization

Parameter	Value
N	75
R	1KΩ
C	10nF
C_3	1nF
K_v	40MHz/V

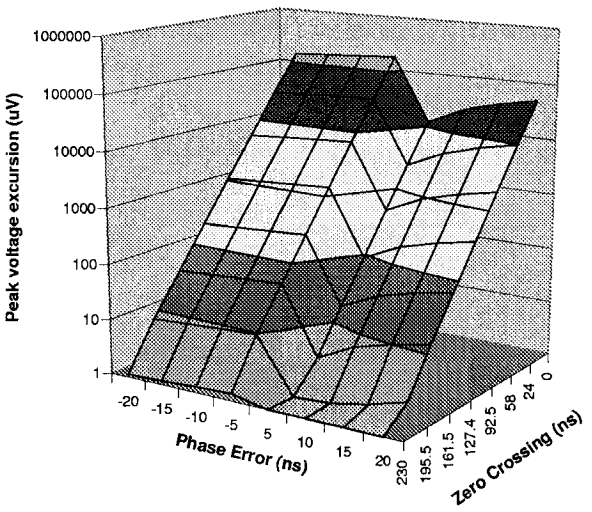


Figure 2-17. Variation of lock time and peak voltage excursion with phase error

Figure 2-17 shows the variation of lock time and peak voltage excursion with phase error. In this experiment, the control voltage is assumed to have an initial value equal to its final value. The phase error is defined as "R-V". This means that a negative phase error implies that the reference signal, R, lags behind the feedback signal, V. During phase locking, the control voltage oscillates around the final value of the control voltage, with the amplitude of each period being smaller than the previous oscillation period. This is because the system used is an underdamped system. The peak voltage excursion is defined as the maximum amplitude between two zero crossings of the final control voltage. The figure reveals three interesting facts; the first two are consistent with second order linear systems, the third one is not. First the zero crossings occur at regular time intervals. This time interval is directly related to the PLL parameters. For smaller values of the damping coefficient, the time between zero crossings shortens. Another interesting phenomenon is that the peak amplitude excursions decrease exponentially with each zero crossing interval. In the case shown in Figure 2-17, the peak amplitude excursions decrease by a factor of five after every zero crossing. Again, this factor is directly related to the loop parameters. The amplitude reduction rate is higher for larger damping coefficients.

The third interesting phenomenon is the variation of the lock time with the polarity of the phase error. As shown in Figure 2-17, positive phase errors lead to substantially faster lock time that negative phase errors. The reason for this is that negative phase errors are really positive phase errors with a phase error equal to $2\pi - \theta_{error}$, or in the time domain, the phase error can be expressed as $T_{ref} - t_{error}$. For phase errors less than half a period, the negative phase error is really a large positive phase error. This can only be remedied by changes to the phase detector.

One important conclusion from the above experiment is that even if the frequencies are initially aligned, a large amount of time might be needed for phase locking. This is true even for small phase errors. Figure 2-18 shows the variation of frequency error with time for different phase error offsets. As shown the lock time does not vary a great deal for phase errors between 5ns up to 20ns.

Figure 2-19 shows the results of another experiment. In this experiment both the phase error and the initial control voltage are varied and the lock time is measured. As the figure reveals there are two local minima in the graph. One occurs when the phase error is 3ns and the control voltage error is 5mV, and the other occurs when the phase error is 5ns and the control voltage is 10mV. This seems to indicate that there is an optimum trajectory in which the lock time is minimized. This is achieved by varying both the phase error and initial frequency error (or control voltage).

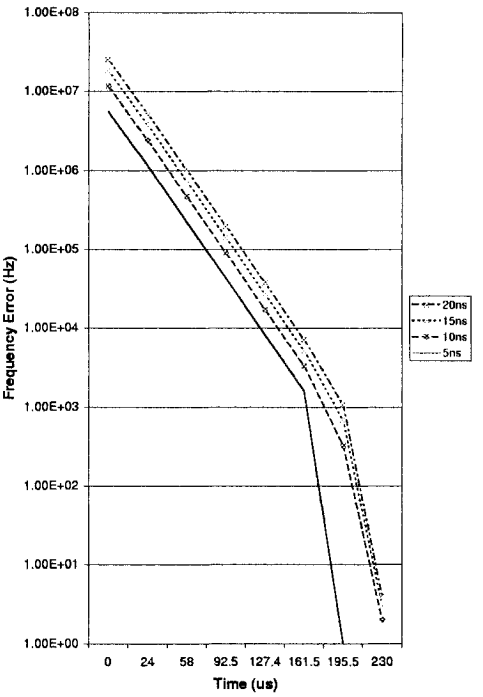


Figure 2-18. Variation of frequency error with time for different phase error offsets

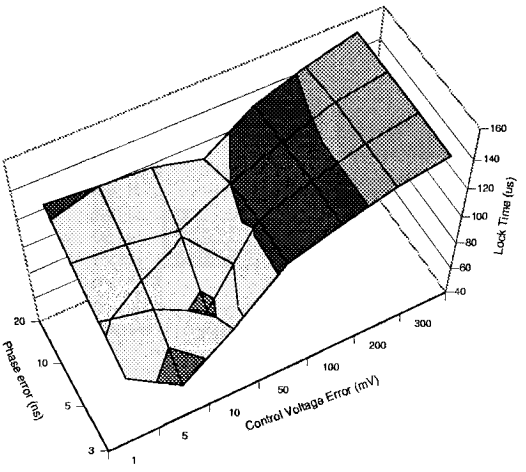


Figure 2-19. Variation of lock time with phase error and initial frequency error

In order to verify this, another simulation was conducted in which the initial output frequency was set to 200KHz away from the targeted frequency. The phase lock was then measured at different phase error offsets. Figure 2-20 shows the result of this experiment. As shown there is a local minimum when the phase error is 2.7ns. At this phase error, the lock time is 38.4μs. This constitutes more than a 2x reduction in phase lock time as compared to a 2ns phase error.

Figure 2-20. Variation of lock time with phase error for initial frequency error of 200KHz

2.6 Summary

In this chapter, PLL basics have been covered. PLLs are an indirect method of frequency synthesis. A low-frequency high-resolution crystal oscillator is used as a reference to tune an on-chip low-cost oscillator, embedded in a feedback closed loop system, to synthesize high frequencies with high precision. The most common implementation of PLLs is the

charge-pump based PLL architecture. In essence, the PLL is a discrete-time nonlinear sampled system. In order to simplify its analysis, the PLL is linearized by assuming it is locked or close to lock condition. Furthermore, if the closed loop bandwidth is at least an order of magnitude less than the sampling period (PFD comparison rate), the PLL can be treated as a linear system. Using this linear analysis, loop parameters such as loop bandwidth can be easily computed. For fast lock PLLs, the nonlinear locking mechanism is important. It was found that lock time is a function of many parameters, including the delay in the PFD's reset circuitry, initial phase error and trajectory of the PLL's system, differences in the initial and target frequencies.

REFERENCES

[1] S. Weber, "Part 1: Mobile Computing – Computer Design," *EE Times*, Issue: 889, Feb 19, 1996.

[2] F. Gardner, "Charge-pump phase-lock loops," *IEEE Transactions on Communications*, vol. COM-28, pp. 1849-1858, Nov. 1980.

[3] D. Wolaver, *Phase-Locked Loop Circuit Design*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1991.

[4] R. Best, *Phase-locked Loops: Theory, Design, and Applications*, New York, McGraw-Hill Inc., New York, 1993.

[5] B. Razavi, "A Study of Phase Noise in CMOS Oscillators," *J. of Solid-State Circuits*, pp. 331-343, March 1996.

[6] C. Vaucher, *Architectures for RF Frequency Synthesizers*, Boston: Kluwer Academic Publishers, 2002.

[7] F. M. Gardner, *Phase-Lock Techniques*, New York: John Wiley & Sons, Inc. 1979.

[8] O. Jacobs, *Introduction to Control Theory*, New York: Oxford University Press, 1993.

[9] J. Hein and J. Scott, "z-Domain Model for Discrete-Time PLL's," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 11, pp. 1393-1400, November 1988.

[10] A. Oppenheim and R. Schafer, *Discrete-time Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1989.

[11] A. M. Fahim and M. I. Elmasry, "A Fast Lock Digital Phase-Locked Loop Architecture for Wireless Applications," *IEEE Transactions on Circuits and Systems II*, pp. 63-72, Feb. 2003.

[12] D. Hess, "Cycle Slipping in a First-Order Phase-Locked Loop," *IEEE Transactions on Communication Technology*, vol. Com-16, no. 2, pp. 255-260, April 1968.

[13] W. Egan, *Frequency Synthesis by phase lock*, Malabar, FL: R.E. Krieger Pub. Co., 1990.

[14] M. Soyuer and R. Meyer, "Frequency Limitations of a Conventional Phase-Frequency Detector," *IEEE J. of Solid-State Circuits*, vol. 25, no. 4, pp. 1019-1022, August 1990.

Chapter 3

LOW-VOLTAGE ANALOG CMOS DESIGN

3.1 Introduction

High-performance and highly integrated SoCs have been mainly fueled by growing demand for high performance and equally growing demand for low cost. Innovations in CMOS technology have made these two seemingly contradicting objectives possible by shrinking the feature length of MOSFET devices. This, however, also leads to the need to operate and low power supplies in order to maintain device reliability. For digital CMOS circuits, this would lead to lower performance, but also lower power consumption. For analog CMOS circuits, this would lead to lower performance, but not necessarily lower power consumption. In this chapter, low-voltage analog CMOS circuit techniques pertaining to PLL design are detailed.

3.2 MOS Transistors

Before discussing low-voltage CMOS analog circuits, some MOS transistor basics are in order. In this section, the basic operation of MOS transistors is reviewed. This is followed by the small-signal derivation of MOS transistors, followed by a review of MOS device noise theory.

MOS Basics. Metal-oxide semiconductor (MOS) devices come in two flavors, PMOS and NMOS devices. Technologies that support both types of MOS devices are called complementary metal-oxide (CMOS) technologies. A cross section of a NMOS device is shown in Figure 3-1. An NMOS device is composed of a source (S), drain (D), and a gate (G) region between

the source and drain. The distance between the source and the drain is often referred to as the length (L) of the NMOS device. The depth dimension of the gate is called the width (W) of the NMOS device. A field oxide insulator (SiO_2) region around the NMOS isolates it from nearby devices.

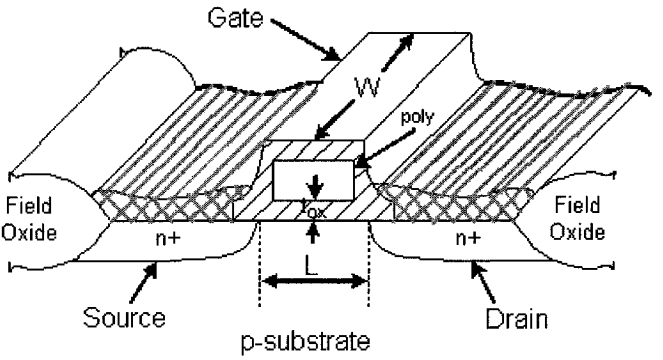
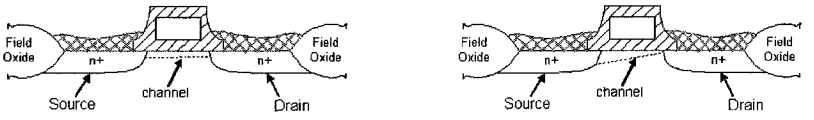


Figure 3-1: Cross section of an NMOS device

The gate acts as a switch controlling the flow of electrons between the source and the drain. The gate terminal and the surface of the p- substrate under the gate effectively act as terminals of a capacitor around the gate's insulating material. When the gate voltage, V_G is increased from 0V to beyond the threshold voltage of the device (V_T), an electric field forms between the terminals of this capacitor and an electric field and negatively charge particles are attracted near the surface of the substrate. This induces a negatively charged channel between the source and drain region, and hence forms a conduction path between the source and drain, as shown in Figure 3-2 (a).



(a) (b)

Figure 3-2: Channel in NMOS device for (a) uniform channel (b) non-uniform channel

The threshold voltage, V_T, of a MOS device is defined as the required gate voltage such that a channel forms between the source and drain. A common expression for the threshold voltage is [1]

$$V_T = V_{T0} + \gamma\left(\sqrt{V_{SB} + 2\psi_B} - \sqrt{2\psi_B}\right) \tag{3.1}$$

where ψ_B is a process dependent parameter defined as the difference between the Fermi level and the intrinsic level of silicon [2]. V_{T0} is also a

process dependent parameter and is defined as the threshold voltage with no potential applied on the source or drain terminals. V_{SB} is the source to substrate voltage. The γ factor is also another process dependent parameter.

If the source and drain terminals of a transistor have different potentials (which is usually the case), current flows between the source and drain. This current increases with applied drain voltage. However, as the drain voltage is increased, an electric field develops between the source and drain, which causes the channel to have a non-uniform shape as shown in Figure 3-2(b). When the depth of the channel at the drain reaches zero, the current no longer increases with drain voltage and the transistor is said to be in saturation. The current versus applied drain to source voltage (V_{DS}) for a constant gate to source voltage (V_{GS}) for an NMOS device is shown in Figure 3-3.

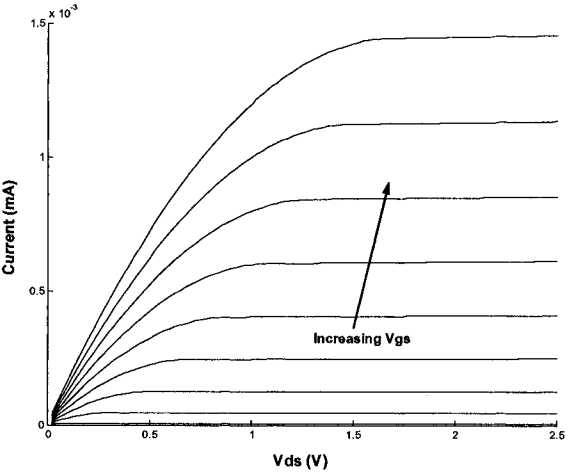


Figure 3-3: I_D versus V_{DS} for NMOS device

CMOS operation is based on charging and discharging a current through the gate capacitance[1]. The gate capacitance is given as

$$C_{gs} = WLC_{ox} \tag{3.2}$$

where C_{ox} is given as

$$C_{ox} = \frac{\varepsilon_0 \varepsilon_{Si}}{t_{ox}} \tag{3.3}$$

where ε_0 is the permittivity of free space (approximately 8.854×10^{-11} F/m), ε_{Si} is the permittivity of silicon dioxide (approximately 3.9), and t_{ox} is the

[1] In general, other capaictances may be present, but for clarity, only gate capacitance of a MOS transistor is considered here.

thickness of the gate oxide material. The units of C_{ox} are hence F/m^2, and C_{gs} is in Farads. When a transistor is in saturation (the case shown in Figure 3-2b), the effective thickness of the capacitor formed across the gate of the MOS device is reduced and is approximated to be 2/3 of its original value [3]. Therefore, the gate to source capacitance in saturation region is given as

$$C_{gs} = \frac{2}{3} WLC_{ox} \tag{3.4}$$

By noting that

$$q = CV \tag{3.5}$$

and integrating by V_{DS} to obtain the current though the drain, it can be shown [4] that the drain current can be expressed as

$$I_D = \mu C_{ox} \frac{W}{L} (V_{GS} - V_T) V_{DS} \tag{3.6}$$

where μ is the mobility of the device. This expression can be further broken down into two forms, depending on the region of operation

$$I_D = \frac{1}{2} \mu C_{ox} \frac{W}{L} (V_{GS} - V_T)^2 \quad , V_{DS} > V_{GS} - V_T \tag{3.7}$$

$$I_D = \mu C_{ox} \frac{W}{L} \left[(V_{GS} - V_T) V_{DS} - \frac{V_{DS}^2}{2} \right] \quad , V_{DS} < V_{GS} - V_T \tag{3.8}$$

where (3.7) is the drain current in saturation region and (3.8) is the drain current in the linear (or triode) region. The quantity V_{GS}-V_T is often referred to as the saturation voltage, $V_{ds,sat}$. Figure 3-4 shows the drain current (I_D) as a function of V_{GS} for both NMOS and PMOS devices. As the figure shows, the transfer function of a PMOS device is the mirror image of that of an NMOS device.

One important effect in MOS devices is the channel modulation effect. Ideally, the drain current should remain independent of drain voltage in the saturation region, as explained above. However, as shown in Figure 3-4, there is a small dependence of drain current on drain voltage. This is explained by the presence of an electric field between the source and drain terminals of a MOS device. This electric field forms a depletion region that grows with the applied voltage, and hence reduces the effective length of a MOS device. This effect is called channel modulation effect and can be modeled by modifying equation (3.7) to

$$I_D = \frac{1}{2} \mu C_{ox} \frac{W}{L} (V_{GS} - V_T)^2 (1 + \lambda V_{DS}) \tag{3.9}$$

where λ is called the *early effect* and has units of V^{-1} and is inversely proportional to the transistor's length, L.

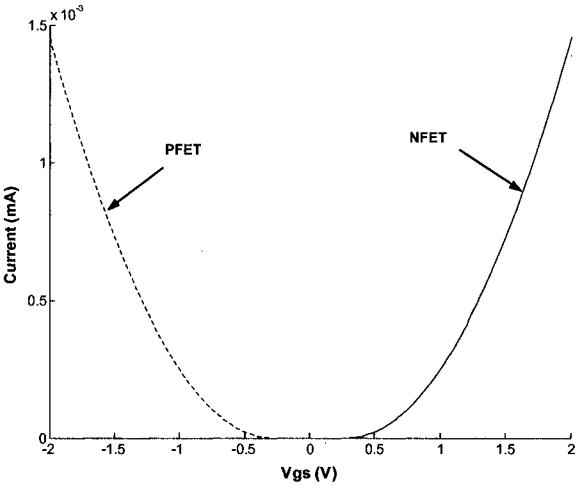


Figure 3-4: I_D versus V_{GS} for both NMOS and PMOS devices

Small-signal Operation. An important parameter in analog CMOS design is the transconductance of a MOS device. Transconductance is defined as

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \tag{3.10}$$

and is defined for a transistor operating in the saturation region. Using equation (3.6), transconductance can be expressed as

$$g_m = \mu C_{ox} \frac{W}{L}(V_{GS} - V_T) \tag{3.11}$$

In analog CMOS design, current is usually fixed by a current source or current mirror. Relating g_m to the drain current I_D using (3.10) and (3.7) yields

$$g_m = \sqrt{2\mu C_{ox} \frac{W}{L} I_D} \tag{3.12}$$

In small signal analysis, it is assumed that DC bias voltages are fixed and a small input signal is applied to the circuit in such that the operating region of the circuit is not changed and circuit nonlinearities have little effect on circuit operation. A simplified small signal equivalent model for a MOS

device is shown in Figure 3-5. The r_{ds} parameter is the DC output resistance, which is given by

$$r_{ds} = \frac{1}{\lambda I_D} \tag{3.13}$$

For DC analysis, all capacitors can be considered as open circuits.

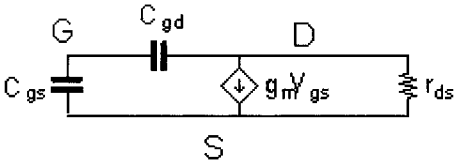


Figure 3-5: Small-signal equivalent model of a MOS device

The unity gain frequency, f_T, of a MOS device is of particular importance in high-frequency design. By applying a current source at the gate input of a MOS device and using the model shown in Figure 3-5 and assuming that $g_m >> f_T C_{gd}$, it can be easily shown that the unity gain frequency

$$f_T = \frac{1}{2\pi} \frac{g_m}{C_{gs} + C_{gd}} \tag{3.14}$$

where the units of f_T is hertz.

Noise in MOS Devices. There are many mechanisms for noise. The two most important in CMOS analog integrated circuits is thermal noise and flicker noise. Thermal noise is due to the random fluctuations of energetic charge carriers in a semiconductor. As temperature increases, this random fluctuation also increases. Thermal noise is white noise, meaning it has a constant spectral density for any given frequency. The spectral density of the voltage noise of a resistor can be given as [5]

$$V_R^2(f) = 4kTR \tag{3.15}$$

where k is Boltzmann's constant (1.38×10^{-23} J/K), T is the temperature in Kelvins, and R is the resistance. Alternatively, the current noise power spectral density can be expressed as

$$I_R^2(f) = \frac{4kT}{R} \tag{3.16}$$

When a MOS transistor is in the linear region, the noise current spectral

density can be derived using (3.16) and (3.8) and is given as

$$I_d^2(f) = 4kT\left[\mu C_{ox}\frac{W}{L}\left[(V_{GS}-V_T)-\tfrac{1}{2}V_{DS}\right]\right] \tag{3.17}$$

For MOS transistors in saturation region, the noise is found by integrating along the nonuniform transistor channel. In general, the noise current spectral density of a MOS transistor in saturation is given by [5]

$$I_d^2(f) = 4kT\gamma g_m \tag{3.18}$$

where γ is a unitless constant that depends on the shape of the channel in saturation for a given value of V_{DS}. For long channel devices, γ is usually given as 2/3. For short channel devices, it can be as high as 2 or 3.

Another important noise mechanism in CMOS devices is flicker noise. Flicker noise is often explained as being caused by charge trapping in the gate of a MOS device due to device imperfections. For a constant applied voltage, the magnitude of noise increases due to the accumulation of charges being trapped in the gate. When the input of the gate is toggled, the charges are cleared and hence noise is reduced. This means that flicker noise is frequency dependent and decreases with frequency. Frequency dependence of flicker noise is modeled as 1/f noise and is given as [5]

$$V_g^2(f) = \frac{K}{WLC_{ox}f} \tag{3.19}$$

where K is a process dependent term and has units of C^2/m^2. Referring this noise to the output yields the flicker noise current density and is given as

$$I_d^2(f) = \frac{K}{f}\frac{g_m^2}{WLC_{ox}^2} \tag{3.20}$$

3.3 Low-Voltage Current Mirrors

One of the most fundamental building blocks of analog integrated circuits is the current mirror. Current mirrors enable a single current source to supply a stable current to several circuit blocks. Two important metrics for current mirrors are output impedance and voltage headroom. The output impedance determines the variation of the mirrored current when the applied voltage varies. Higher output impedance implies less current variation with applied voltage, and hence a more stable current source. Voltage headroom specifies how much voltage drop across the current mirror is required to

operate the current mirror reliably. This is especially important for low-voltage circuit design.

The simplest current mirror is shown in Figure 3-6. An external current source is supplied to transistor MN1, which has its gate and drain terminals short-circuited (diode connected). Since the V_{DS1} of transistor MN1 is equal to V_{GS1}, it is in saturation. Since both the gate terminals and source terminals of MN1 and MN2 have the same potential, it follows from equation (3.7) that I_{in} is equal to I_{out} (some variation in current will exist due to channel modulation effect). However, due to the I_D dependence on V_{DS}, I_{out} is not necessarily constant. This variation is governed by the output impedance of MN2, which is given by equation (3.13). Since the required voltage headroom is only one transistor in saturation, the voltage headroom then becomes a V_{GS}-V_T, or $V_{ds,sat}$.

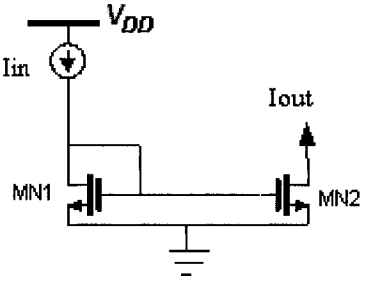


Figure 3-6: Simple current mirror

In order to obtain better matching between the input and output currents, higher output impedance is needed. One commonly used implementation is the cascode current mirror shown in Figure 3-7. In effect, the original current mirror pair MN1 and MN2 is used as the bias devices, and transistors MN3 and MN4, called cascode devices, increase the output impedance. In order to analyze the operation of this circuit, the small signal equivalent circuit is shown in Figure 3-8. Notice that the equivalent circuit transistor MN2 is simply r_{ds2}. The reason for this is that the both the gate and source voltage are at AC ground, and hence the small signal voltage controlled current source is off. In order to find the output impedance, a voltage source v_x is applied to the output and the resulting current i_x is computed. The output impedance is then given as v_x/i_x. The current i_x can be given as

$$i_x = g_{m4} v_{gs4} + \frac{v_x - v_s}{r_{ds4}} \tag{3.21}$$

and

$$v_s = i_x r_{ds2} \tag{3.22}$$

therefore,

$$i_x = g_{m4}\left(-i_x r_{ds2}\right) + \frac{v_x - i_x r_{ds2}}{r_{ds4}} \tag{3.23}$$

rearranging for v_x/i_x results in the output impedance and is given as

$$r_{out} = r_{ds2} + r_{ds4} + g_{m4} r_{ds2} r_{ds4} \approx g_{m4} r_{ds2} r_{ds4} \tag{3.24}$$

In order to operate the cascode current mirror correctly, all transistors MN1-MN4 must be in saturation. This is ensured by the diode connected transistors MN1 and MN3, which implies that the required output voltage headroom for the cascode current mirror is $V_{T2}+V_{T4}$.

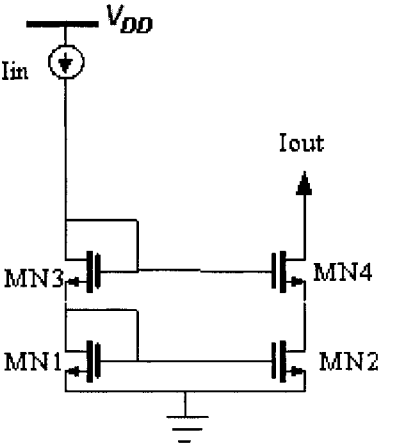


Figure 3-7: Cascode current mirror

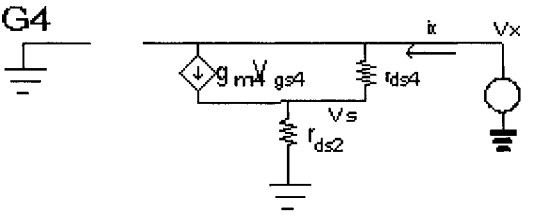


Figure 3-8: Small-signal equivalent circuit of cascode current mirror

In order to reduce the required headroom for a cascode current mirror, a wide swing cascode current mirror shown in Figure 3-9 can be used [4]. The relative sizes shown in Figure 3-9 show that the cascode devices are sized n^2 times smaller and are biased by a fifth device, MN5. To find the required output voltage headroom, v_{ds2} is first computed by noting that

$$v_{ds2} = v_{G5} - v_{GS4} \tag{3.25}$$

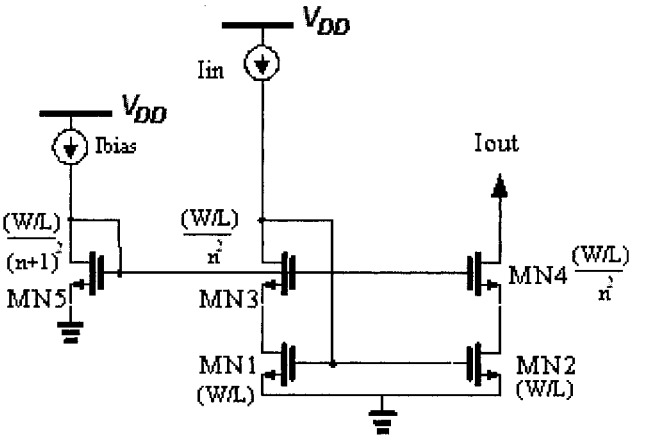


Figure 3-9: Wide-swing cascode current mirror.

To find the minimum required voltage headroom, let $v_{ds,sat} = v_{gs}-v_T$ and from (3.25) we have

$$v_{ds2} = (v_{G5} - v_T) - (v_{GS4} - v_T) \tag{3.26}$$

$$v_{ds2} = v_{ds,sat5} - v_{ds,sat4} \tag{3.27}$$

$$v_{ds2} = (n+1)v_{ds,sat5} - nv_{ds,sat4} = v_{ds,sat2} \tag{3.28}$$

Also, the minimum voltage required to keep transistor MN4 in saturation is $n \cdot v_{ds,sat2}$. Using this result and (3.28), the minimum required voltage head for a wide-swing cascode current mirror is $(n+1) \cdot v_{ds,sat2}$. For minimum voltage headroom, set n=1, and the minimum voltage headroom becomes $2 \cdot v_{ds,sat2}$. This is nearly double of that of a regular current mirror; however, it has the two advantages of having lower voltage headroom than a regular cascode and maintaining the output impedance of a regular cascode. The output impedance is the same as a regular cascode current mirror.

In some cases, a better trade-off between output impedance and voltage headroom is required. A pseudo-cascode [6], shown in Figure 3-10, is capable to providing this trade-off. This current mirror is also known as a self-cascode current mirror [7]. Using a small signal AC analysis, the output impedance can be found to be

$$r_{out} \cong \frac{g_{m3} r_{ds3}}{g_{m2} + 1/r_{ds2}} \tag{3.29}$$

As $g_{m3}/g_{m2} \rightarrow \infty$, the output impedance approaches to that of a regular cascode and the output voltage headroom stays as $2 \cdot v_{ds,sat}$. As $g_{m3}/g_{m2} \rightarrow 0$, then transistor MN3 will be in triode with very low output impedance (large g_{m3} implies large W/L ratio, and hence small r_{ds3}) and hence the output impedance of the pseudo cascode will be dominated r_{ds2}. The output voltage headroom will also be dominated by MN2 and will be $v_{ds,sat}$.

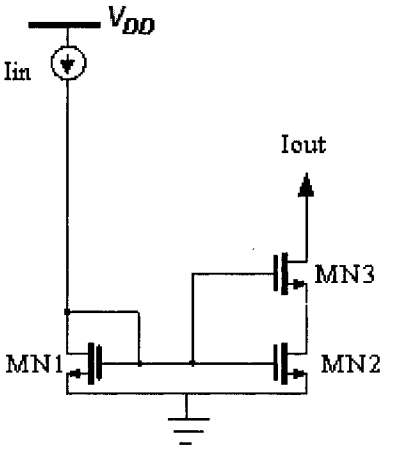


Figure 3-10. Schematic diagram of a pseudo cascode

The different current mirrors are summarized in Table 3-1. As the table shows, the simplest current mirror also requires the smallest output voltage headroom making it most suitable for low-voltage applications. Its disadvantage is low output impedance. The wide-swing cascode gives excellent output impedance, but requires twice as much output voltage headroom as a simple current mirror.

Table 3-1. Summary of current mirrors

	Output Impedance	**Output Voltage Headroom**
Simple current mirror	$r_{ds} = 1/(\lambda I_D)$	$V_{ds,sat}$
Cascode current mirror	$g_{m4}r_{ds4}r_{ds2}$	$2V_T$
Wide-swing cascode	$g_{m4}r_{ds4}r_{ds2}$	$2 \cdot V_{ds,sat}$
Pseudo cascode	r_{ds} to $g_{m4}r_{ds4}r_{ds2}$	$V_{ds,sat}$ to $2 \cdot V_{ds,sat}$

3.4 Low-Voltage Charge Pumps

The charge pump is responsible for raising or lowering the loop filter voltage so as to tune the VCO to a desired frequency. As shown in Figure 3-11, a charge pump consists of two current sources with control switches gating the flow of current into or out of the loop filter. I_{UP} and I_{DN} are designed to be equal to the charge pump current, I_p. The amount of voltage needed for both the UP and DN currents eats away from the maximum voltage swing across the loop filter voltage, $V_{FILT,max}$. As the supply voltage is scaled, $V_{FILT,max}$ decreases proportionally. In order to be able to tune to the desired frequency rang with reduced $V_{FILT,max}$, the VCO gain, K_V, must increase. This, however, increases the VCO's sensitivity to noise injected onto the loop filter and increases the PLL's overall jitter. For this reason, it is important to minimize the required voltage headroom of the UP and DN current sources of the charge pump.

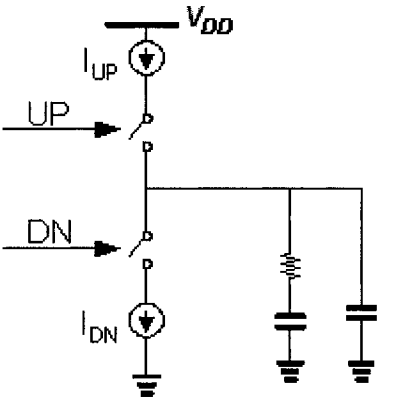


Figure 3-11. Simplified charge-pump diagram

Another issue associated with charge pumps is how to implement the switches. The simplest implementation of the switches would be to implement the UP switch with a PMOS and the DN switch with an NMOS. This, however, would mean that the current sources would be turned off when the switch is open (no DC path to ground or supply cuts off the current from the current source FETs). When the UP or DN switch is closed again, the current source would need some time before settling to the correct value. This would cause severe variation in charge pump current, and hence loop filter dynamics. Jitter due to this effect would manifest itself as deterministic jitter that varies with process and temperature. This effect is shown in Figure 3-12.

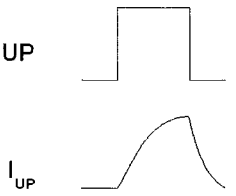


Figure 3-12. Effect of turning current source on/off on charge pump current output

To alleviate this effect, a current steering charge pump, shown in Figure 3-13, is commonly used. The current sources in this configuration are never turned off. This charge pump consists of four switches, the UP and DN current sources, and a unity gain buffer. Consider the case when UP=DN=0. In this case, switches 1 and 3 are closed, and switches 2 and 4 are open. The loop filter is tristated as desired. The I_{UP} is flows through switches 1 and 3 and is sunk by the DN current source. In the case UP=1 and DN=0, switches 2 and 3 are closed, and switches 1 and 4 are open. In this case, I_{UP} flows through switch 2 and charges up the loop filter, as desired. The I_{DN} current is sourced by the unity gain buffer through switch 3. In the opposite case of UP=0 and DN=1, switches 1 and 4 are closed, and switches 2 and 3 are open. In this case, the I_{UP} current is sunk by the unity gain buffer through switch 1 and the I_{DN} current draws current away from the loop filter. In the last case when UP=DN=1, switches 2 and 4 are closed, and switches 1 and 3 are open. In this case I_{UP} current is sunk by the DN current source through switches 2 and 4.

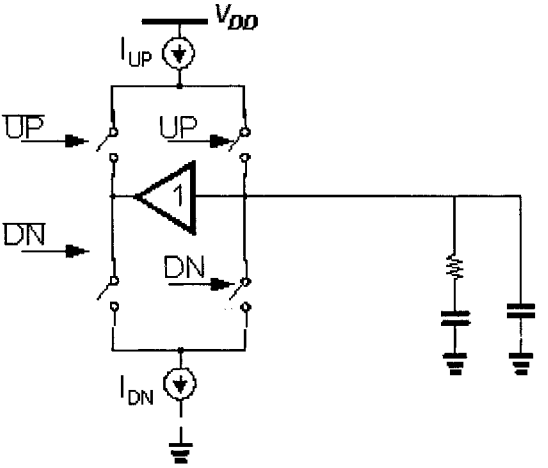


Figure 3-13. Current steering charge pump

For 1.5V operation and lower, different topologies are necessary. Take

the case when wide-swing cascode devices are used in both UP and DN current sources. For a typical value of $v_{ds,sat}$ of 300mV is used, the current sources would require 1.2V of total voltage headroom, leaving only 300mV for $V_{FILT,max}$. This is unacceptable for most applications. In order to gain more headroom, a simple current mirror may be the only solution. This would give a $V_{FILT,max}$ of 900mV, which is nearly only half of the supply voltage. If more voltage headroom is required, a different charge pump topology is required. Furthermore, matching between UP and DN currents is poor and may result in a PLL with overall large deterministic jitter.

Figure 3-14 shows a schematic of a low-voltage charge pump [9]. A simple current mirror is used for the I_{DN} current and a feedback mechanism enabling operation to V_{DD}. The circuit operation is best described through and example. Node F is the maximum voltage that the PMOS current mirror can be while still remaining in saturation. When the loop filter voltage (node E) goes higher than node F, the PMOS current starts to become smaller the NMOS current; however, the amplifier lowers the gate voltage of the UP current mirror such that the UP current matches the DN current. This mechanism is similar to the current control in class AB buffer reported in [10]. The total voltage headroom required is only one $v_{ds,sat}$.

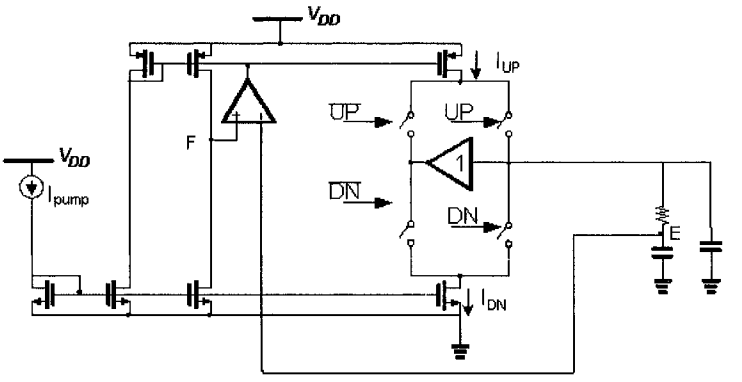


Figure 3-14. Low-voltage charge pump schematic

3.5 Low-Voltage Oscillator Design

The voltage-controlled oscillator (VCO) is responsible for synthesizing the correct output frequency. As discussed in section 3.4, it is essential to maintain the maximum possible $V_{FILT,max}$ in order to maintain a relatively low VCO gain, K_v. A low K_v reduces the VCO's sensitivity to noise injected into the loop filter from external sources (such as supply bounces, noise coupling through substrate, nearby high frequency lines, etc.). The noise

performance of VCOs will be studied in detail in chapter 4. In order to cover a certain frequency range, the VCO gain must be sufficiently large to cover all frequencies over all corners. In other words,

$$K_v \geq \frac{f_{out,max} - f_{out,min}}{V_{filt,max}} \tag{3.31}$$

where $f_{out,max}$ is the maximum required output frequency, $f_{out,min}$ is the required minimum output frequency. The output range of PLL for SoC processors is usually quite large since it is desirable to use the PLL for multiple SoC cores. As the supply voltage is scaled, the VCO gain must be increased, which inevitably leads to larger jitter. In this section, a solution to this problem is given as well as analyzing the low-voltage capability of some commonly used oscillators.

3.5.1 Digital Tuning Circuitry

One commonly used strategy for low-voltage and high output frequency range requirements is to use a digital tuning circuitry [11]. The PLL locking procedure is broken into two steps: coarse tuning and fine tuning. Coarse tuning is performed by digitally controlling the frequency of the VCO using a digital control loop. Once the VCO is brought close to the desired frequency, the regular analog control PLL loop takes over. The topology of this two-loop architecture is shown in Figure 3-15. The advantage of this technique is that the VCO gain is no longer required to be large enough to synthesize all desired frequencies given a $V_{FILT,max}$. Instead, the VCO gain only needs to be sufficiently large to be able to synthesize a frequency range spanning on coarse tune setting. During coarse tuning, the loop filter voltage is held to a constant value.

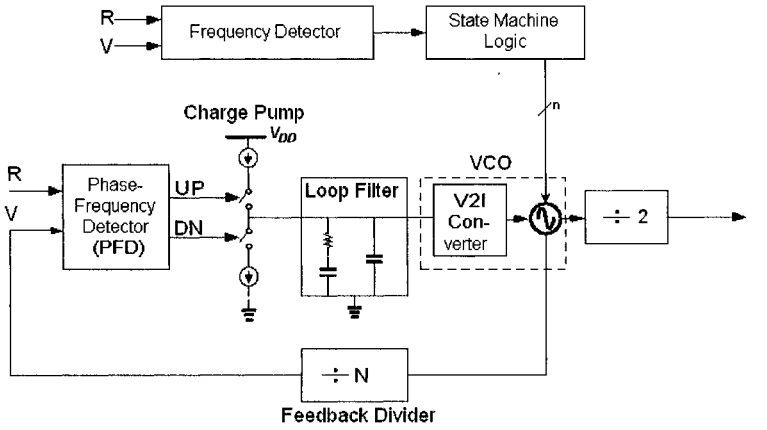


Figure 3-15. PLL topology with coarse tuning control loop

The VCO frequency transfer function is shown graphically in Figure 3-16. As the figure shows there is some overlap in output frequency between adjacent coarse tune settings. This is necessary in order to guarantee that there are no gaps in frequency that the VCO cannot synthesize. This is necessary since the VCO gain as well as its center frequency change over process and temperature. As the figure shows, the VCO gain even varies with output frequency. In this particular example, the VCO gain decreases with frequency. In other VCO topologies (especially LC VCOs), the opposite may be true.

A state machine and a frequency detector determines how the VCO is controlled digitally. In Figure 3-17, a binary algorithm is shown. The algorithm starts with a code in the middle of the tuning range. A frequency detector counts the number of VCO cycles per reference period. This may implemented as two counters that count a fixed number of VCO cycles and reference cycles. If the VCO counter finishes first, then the VCO frequency is too high, and a logic '0' is assigned to the MSB. The procedure is repeated for the 2nd MSB, followed by the 3rd MSB, until the LSB is reached.

Two important observations must be made. First, the digital tuning algorithm requires

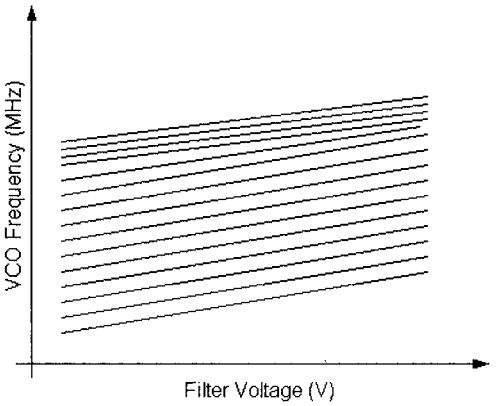


Figure 3-16. VCO characteristics with digital coarse tuning

$$T_{lock} = b \cdot T_{ref} \tag{3.32}$$

time to settle to the correct value, where b is the number of bits used to coarse tune the VCO and T_{ref} is the reference period. After this period, the regular analog PLL loop takes over and requires additional time to lock. Also, the final coarse tune setting may be off by 1 LSB. This means that a minimum overlap of two coarse tune settings is required in the VCO characteristic shown in Figure 3-16. Additional overlap is also necessary to

cover temperature and process variation. This increases the minimum VCO gain requirement to

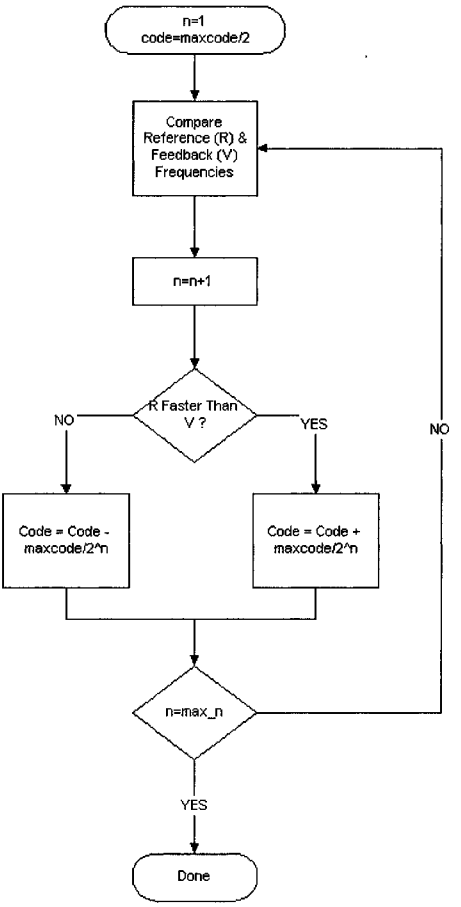


Figure 3-17. State flow machine of binary locking algorithm

$$K_{v,min} = \frac{\Delta F}{V_{filt,max}} \cdot \frac{1}{2^{n-1}} \cdot P \tag{3.33}$$

where ΔF is the required output frequency range and P is a factor to over process and temperature variation.

3.5.2 Ring Oscillator Design

The most commonly used type of VCO used for SoC processors is a ring oscillator. A ring oscillator-based VCO is composed of two blocks: the voltage-to-current converter (V2I) and the current controlled oscillator (ICO) as shown in Figure 3-18. There are two reasons to partition the VCO in this

manner. Firstly, controlling VCO gain and output frequency are now decoupled. The VCO gain can be set by controlling the V2I gain, whereas the ICO can be optimized for low-jitter and low-power consumption. Secondly, this type of configuration enables a linear voltage-to-frequency relationship and hence ideally enabling a fairly constant VCO gain.

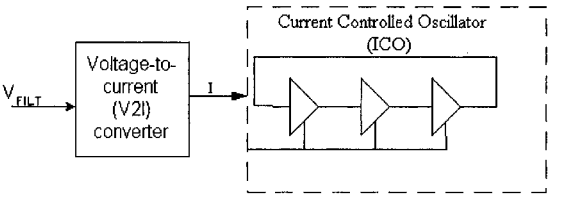


Figure 3-18. Typical VCO used in SoC Processors

This can be shown mathematically by considering the following analysis. A single transistor can be used as a V2I converter. The gate of the transistor is connected to the loop filter voltage and the drain is the output of the V2I converter. In this case, the output current is

$$I_{V2I} = \frac{1}{2}\mu C_{ox}\frac{W}{L}(V_{FILT} - V_T)^2 \tag{3.34}$$

Figure 3-18 shows a three stage ring oscillator. In general, one can have an N stage ring oscillator. In this case, the output frequency of the ring oscillator is given as

$$f_{osc} = \frac{I_{SS}}{2 \cdot N \cdot C_L \cdot V_{SW}} \tag{3.35}$$

where I_{SS} is the bias current from the V2I converter, C_L is the parasitic load capacitance at the output of each stage in the ring oscillator, and V_{SW} is the voltage swing output of each stage of the ring oscillator. The voltage swing is usually given as $I_{SS} \cdot R_{LOAD}$, where R_{LOAD} is the resistive load at the output of each stage of the ring oscillator. If a FET device in saturation is used as an active load, V_{SW} becomes proportional to the square root of I_{SS} (from equation 3.7). This means than the output frequency, f_{osc}, can be expressed as

$$f_{osc} \propto \frac{I_{SS}}{V_{SW}} \propto \frac{I_{SS}}{\sqrt{I_{SS}}} = \sqrt{I_{SS}} \propto V_{FILT} \tag{3.36}$$

This means that the output frequency is now a linear function of the input voltage. In practice, however, the VCO transfer function will have some nonlinearity due to the early effect. If a different, more linear, type of

resistive load is used in the ring oscillator, then a linear V2I converter is needed.

Voltage-to-current converter. Just as with the charge pump, the headroom required for current mirrors in the V2I converter reduce $V_{filt,max}$. Figure 3-19 shows a typical V2I converter. As shown the both the NMOS input and the PMOS current mirror devices need to stay in saturation and strong inversion. The required voltage headroom is thus $2 \cdot V_T$. The allowed input voltage headroom, however, is from V_{TN} to V_{DD}.

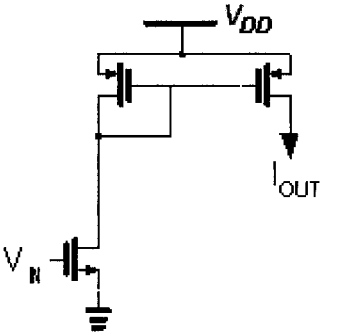


Figure 3-19. A simple V2I converter

The $V_{FILT,max}$ is determined by the required gain and relative sizing of the PMOS and NMOS devices in the input stage of the V2I converter. For the NMOS device to remain in saturation, it is required that $V_{DS,N} > V_{GS,N} - V_{TN}$, where $V_{DS,N}$ is given by

$$V_{DS,N} = V_{DD} - V_{GS,P} = V_{DD} - \left(V_{TP} + \sqrt{\frac{I}{\beta_P}} \right) \tag{3.37}$$

where β is given by

$$\beta = \tfrac{1}{2} \mu C_{ox} \tfrac{W}{L} \tag{3.38}$$

and I is the current resulting from the NMOS transistor given and input filter voltage, V_{FILT}. Therefore, the requirement for saturation then becomes

$$V_{DD} - V_{TP} - \sqrt{\frac{\beta_N}{\beta_P}} \cdot V_{TN} > \sqrt{\frac{\beta_N}{\beta_P}} \cdot V_{FILT,max} \tag{3.39}$$

Rearranging this expression and solving for $V_{FILT,max}$ shows the maximum allowable loop filter voltage to be

$$V_{FILT,max} < \frac{V_{DD}}{\alpha} - \frac{V_{TP}}{\alpha} + V_{TN} \tag{3.40}$$

where α is the ratio

$$\alpha = \sqrt{\frac{\beta_N}{\beta_P}} \tag{3.41}$$

Examining equation (3.40) shows a few interesting facts. First, as $\alpha \rightarrow \infty$, the $V_{FILT,max}$ becomes less than V_{TN}. However, for the NMOS device to stay in saturation and strong inversion, the loop filter voltage must be greater than V_{TN}. Therefore, in this limiting case, there is no headroom available for the loop filter voltage. In the other extreme, as $\alpha \rightarrow 0$, the loop filter voltage headroom extends to infinity, which cannot be physically realizable. Using equation (3.40) to find the requirement on α for V_{FILT} to reach V_{DD}, results in the following constraint:

$$\alpha = \frac{V_{DD} - V_{TP}}{V_{DD} - V_{TN}} \tag{3.42}$$

For $V_{TP}=V_{TN}$, equation (3.25) reduces to $\alpha=1$. Since α is dependent on process parameters and device geometry, it is expected that its value will vary with process and temperature (NMOS & PMOS device variation do not necessarily track each other). If process and temperature variation cause $\alpha>1$, the NMOS device may fall out of saturation. For this reason, α is usually designed to be 2 to 3 to cover process and temperature variation.

Another requirement is that the gain of the V2I converter be linear with the input voltage (for quadratic variation in current). The gain is given by the transconductance of the NMOS device, which is

$$V2I_{gain} = g_{mn} = \frac{\beta_N}{2} \cdot (V_{FILT} - V_{TN}) \tag{3.43}$$

To obtain a required gain, the size of β_N may be relatively large. This means that β_P must also be large. For large gain requirements, the poles introduced by the PMOS current mirrors may start to approach the closed loop bandwidth of the PLL and alter the PLL dynamics. Multiplication in the current mirror may be used to mitigate this, however, this comes at increase device noise and hence jiter.

One way to operate up to the supply voltage while increasing the required α ratio is to use the V2I topology shown in Figure 3-20. The PMOS diode connected transistor of Figure 3-19 is replaced by transistors MP1, MP2, and current source I_B. This type of topology is used for low voltage current

mirrors and amplifiers [12]. The circuit operation is best described through an example. As the filter voltage is increased, the current pulled through MP1 increases, which increases its required V_{GS}. Transistor MP2 is operated in saturation mode for all possible current inputs into MP1. The V_{DS} of transistor MN1, however, is only reduced from the supply by $v_{ds,sat}$ of transistor MP1, which is given as V_{GSP1}-V_{TP1}. In comparison to the previous V2I topology, an extra V_{TP} of voltage has been recovered. For MN1 to stay in saturation, $V_{DSN} > V_{GSN} - V_{TN}$ or more explicitly,

$$V_{DD} - (V_{GSP1} - V_{TP1}) > V_{FILT} - V_{TN} \tag{3.44}$$

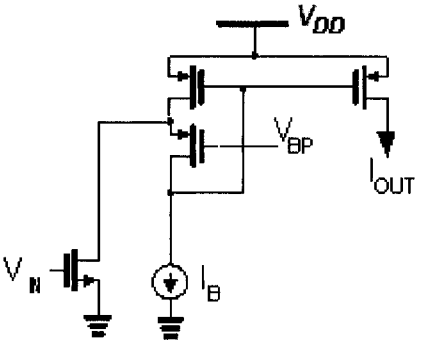


Figure 3-20. A low-voltage V2I converter

Using the same notation for α and β as above and solving for V_{FILT}, equation (3.44) can be rewritten as

$$V_{FILT} < \frac{V_{DD} + V_{TN}(2+\alpha)}{\alpha + 1} \tag{3.45}$$

Setting V_{FILT} to V_{DD} and solving for α results in

$$\alpha < \frac{2V_{TN}}{V_{DD} - V_{TN}} \tag{3.46}$$

For $V_{DD}<3 \cdot V_{TN}$, $\alpha>1$ as desired. However, when larger supply voltages are used, this technique does not offer much of an advantage over the previous V2I topology in terms of increasing the $V_{FILT,max}$

In both V2I topologies discussed above, the filter voltage is limited on the low side by V_{TN}. An alternative method of recovering voltage headroom is to use a rail-to-rail V2I converter [13]. This V2I converter has two input stages. One is a NMOS input stage that works from $V_{DD}/2$ to V_{DD}. The other is a PMOS input stage that works from 0V to $V_{DD}/2$. Only one of two input stages is used at any given time and is selected by a comparator with

hysteresis. The inputs of the comparator are a reference voltage, which sets the selection point between the two input stages, and the loop filter voltage. When the loop filter voltage is higher than the reference voltage, the NMOS input stage is used. When the loop filter voltage is lower than the reference voltage, the PMOS input stage is used. The V2I gain between the two input stages will vary differently with process and temperature, and thus some gain mismatch between the two input stages is inevitable.

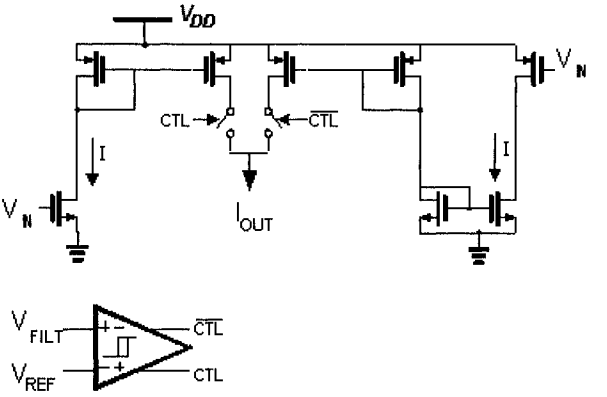


Figure 3-21. Circuit topology for rail-to-rail V2I converter

In order to prevent frequent toggling between the two input stages when the loop filter is close to the reference voltage, hysteresis is built into the comparator. A schematic of such a comparator is shown in Figure 3-22 [16]. In order to obtain hysteresis, transistors MP1 and MP2 must be sized larger than transistors MP3 and MP4. The larger the ratio of the sizes of the two pairs of devices, the more hysteresis is built in to the comparator.

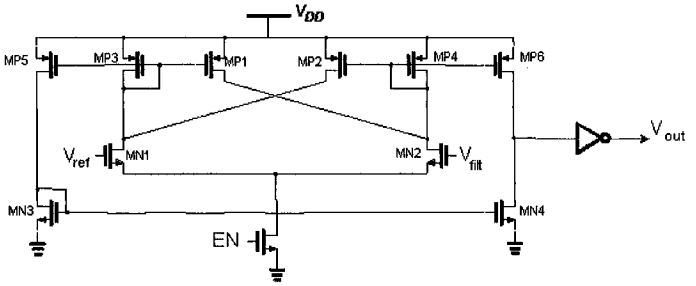


Figure 3-22. Schematic of a comparator with hysteresis

Ring Oscillator Core. There are several methods of designing ring oscillators. Two general methods are considered here: single-ended and differential ring oscillators. A generic single-ended ring oscillator is shown in Figure 3-23. In this case, a 3-stage ring oscillator is shown and the delay stages are simply CMOS inverters. A single current source is used for all

three delay stages. This is done in order to prevent the current source from turning on and off and thus disrupting the value of the current being supplied (as was shown in the case of a single-ended charge pump shown in Figure 3-12). The voltage swing of the ring oscillator is V_{DD}-V_{ISRC}, where V_{ISRC} is the output voltage headroom required for the current source. The ring oscillator's frequency is given as

$$f_{osc} = \frac{I_{SS}}{2 \cdot N \cdot C_L \cdot (V_{DD} - V_{ISRC})} \tag{3.47}$$

The load capacitance is time varying since the PMOS and NMOS devices' region of operation change between linear and saturation as the voltage swing of the delay stages vary. This is due to the gate capacitance's dependence on operating region (see equations 3.2 and 3.4). The minimum voltage required to operate this oscillator is given as

$$V_{DD,min} = V_{ISRC} + V_{SW} \tag{3.48}$$

where V_{ISRC} is the voltage headroom required for the current source and V_{SW} is the required voltage swing, which is defined by the oscillator jitter requirement.

A delay stage of a differential ring oscillator is shown in Figure 3-23. The operating frequency of an N stage differential ring oscillator with PMOS diode loads is given by

$$f_{osc} = \frac{I_{SS}}{2 \cdot N \cdot C_L \cdot \left(V_{TP} + \sqrt{\frac{I_{SS}}{\mu C_{ox} \left(\frac{W}{L}\right)_p}} \right)} \tag{3.49}$$

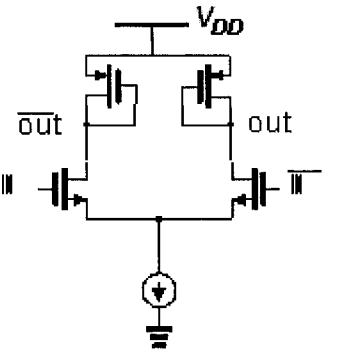


Figure 3-23. Delay stage of a differential ring oscillator

The minimum voltage required voltage for this delay stage is given as

$$V_{DD,min} = V_{SW} + V_{ds,satN} + V_{ISRC} \tag{3.50}$$

where $V_{ds,satN}$ is the saturation voltage of the NMOS input differential pairs. In order for the differential pairs to stay in saturation, the minimum voltage swing is given as

$$\begin{aligned} V_{DS} &> V_{GS} - V_{TN} \\ V_{DD} - V_{SW} - V_{ISRC} &> V_{DD} - V_{TN} - V_{ISRC} \\ V_{SW} &< V_{TN} \end{aligned} \tag{3.51a-c}$$

This places an upper limit on the voltage swing in order for the differential pair to operate correctly. The threshold voltage of the NMOS device is modified by the body effect and hence increases the allowed voltage swing.

Other types of loads can also be used. One such type of load is often referred to as the Maneatis load [14]. An ICO using Maneatis loads is shown in Figure 3-24. The main advantage of the Maneatis load is that is reduces the delay cell's sensitivity to the supply voltage, as shown in Figure 3-25. The disadvantage of this type of load is the increased parasitic capacitance on the delay cell's output, which reduces the maximum achievable voltage swing.

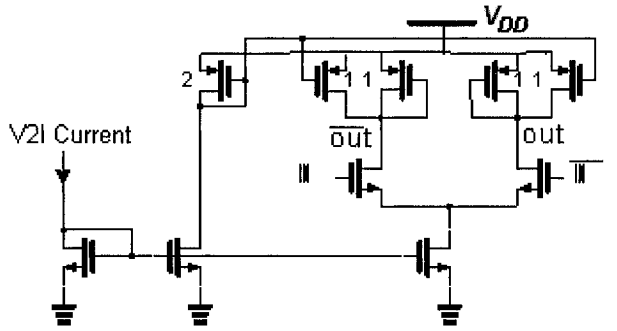


Figure 3-24. ICO topology using Maneatis load

The delay cell's sensitivity to the supply voltage can be completely eliminated by using the topology shown in Figure 3-26. In this case, the voltage swing is referenced by a voltage reference, which is set to V_{DD}-V_{SW} [15]. The external voltage reference can be generated by a bandgap voltage generator (explained in the next section). The PMOS load devices are biased in the linear region. The voltage swing, V_{SW}, is set by setting $V_{REF} = V_{DD} - V_{SW}$. If the voltage swing in the ICO is too high, the feedback error amplifier's output is reduced, which reduces the on resistance of the PMOS devices, and hence the voltage swing is reduced. A similar argument holds for the case when the voltage swing is too small. If the power supply

bounces, the reference voltage, $V_{ref,}$ also bounces by the same amount. Since the V2I current is also assumed to be independent of the power supply, this ICO topology is power supply insensitive.

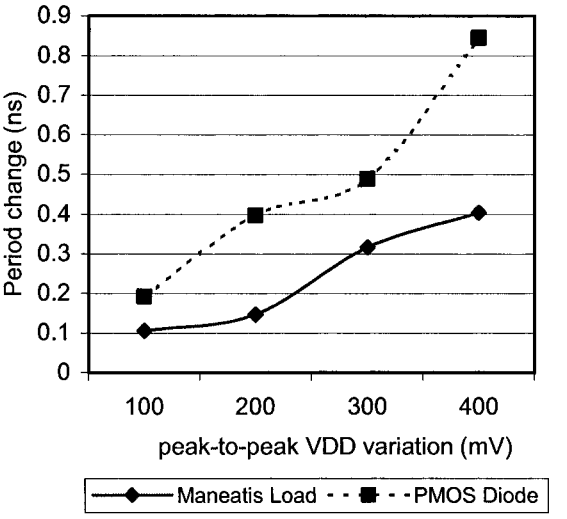


Figure 3-25. Supply sensitivity of Maneatis load versus PMOS diode load

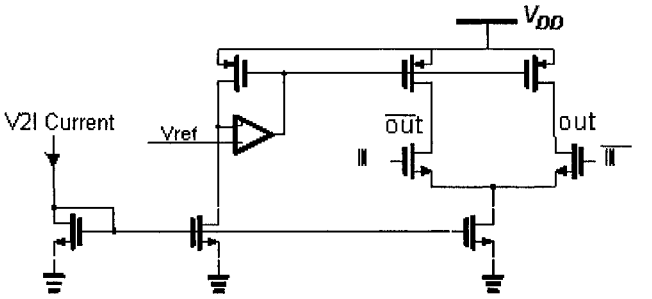


Figure 3-26. Supply voltage insensitive ICO topology

3.5.3 LC VCO Design

In many cases, ring oscillators provide adequate jitter performance and require small area and a reasonable power budget. For some high-performance, low-power applications, however, ring oscillators may not be the best solution. In these cases, LC VCOs may be preferred. An LC VCO makes use of a high-Q LC tank, as shown in Figure 3-27(a). The parasitic resistances of the inductor and capacitor can be modeled as an equivalent

parallel resistor if the Q of the LC tank is large (>10) [17], as shown in Figure 3-27(b).

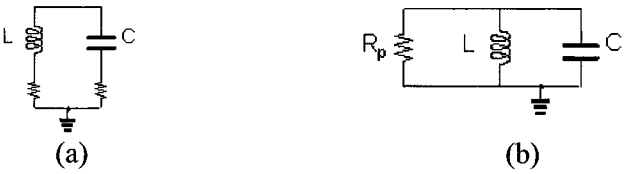


Figure 3-27. LC tank with (a) parasitic resistances in L & C (b) equivalent parallel resistance

The basic idea behind an LC VCO is to provide a negative feedback resistance large enough to cancel out the tank equivalent resistance. An LC VCO commonly implemented in CMOS technologies is shown in Figure 3-28. The negative resistance is provided by the cross coupled NMOS devices. The NMOS devices are sized in such a way that their equivalent g_m is larger than that of the LC tank. The frequency is varied by using a variable capacitor, or a varactor. The loop filter voltage can be used directly as input control voltage to the varactor.

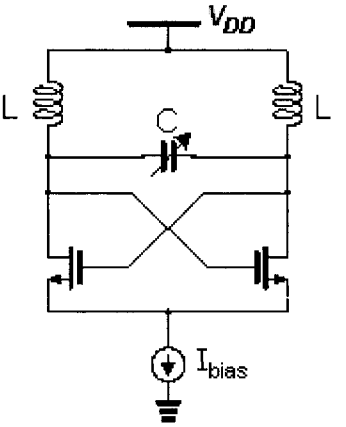


Figure 3-28. Typical CMOS LC VCO

3.6 Voltage and Current References

PLLs for SoC processors need to be fully integrated. This means that stable current and voltage references for the charge pump, V2I, and ICO need to be generated on-chip. In this section, low-voltage current and voltage references that provide good immunity to temperature and process variation are explained.

The core of all absolute voltage references are centered around the bandgap voltage of silicon. The bandgap voltage is the minimum required voltage to turn on a p-n junction in a diode. The current flowing through a diode is a function of this applied voltage and is given as

$$I_D = I_S \cdot \exp\left[\frac{V_D}{V_t}\right] \tag{3.52}$$

where V_D is the voltage applied across the p-n junction diode, I_D is the current through the diode, and $V_t = kT/q$, where k is the Boltzman constant and is the charge of an electron. V_t at room temperature has a voltage of 26mV. I_s is a constant that is temperature dependent and is given as

$$I_S = b \cdot T^{4+m} \exp\left[\frac{-E_g}{kT}\right] \tag{3.53}$$

where b is a proportionality factor, E_g is the bandgap energy of silicon, and m is a constant.

Stability over process is possible since the bandgap voltage depends on the bandgap energy of silicon, which is stable. The voltage across a diode, however, does vary considerably over temperature. In order to explore the temperature dependence of the diode voltage, equation (3.52) is rewritten in terms of V_D and differentiated with respect to temperature. This gives the following expression

$$\frac{\partial V_D}{\partial T} = \frac{\partial V_t}{\partial T} \cdot \ln\left[\frac{I_D}{I_S}\right] - \frac{V_t}{I_S} \cdot \frac{\partial I_S}{\partial T} \tag{3.54}$$

where

$$\frac{V_T}{I_S} \cdot \frac{\partial I_S}{\partial T} = V_t\left(\frac{4+m}{T} + \frac{E_g}{kT^2}\right) \tag{3.55}$$

Substituting (3.55) into (3.54) gives

$$\frac{\partial V_D}{\partial T} = \frac{V_D - V_t(4+m) - E_g/q}{T} \tag{3.56}$$

This shows that the temperature dependence of the diode voltage depends on the absolute values of the diode voltage and temperature itself. Furthermore using typical values for diode voltage and temperature of 750mV and 300°K, respectively, gives a value for the temperature coefficient of -1.5mV/°K to -2mV/°K. This shows that the diode voltage has a negative temperature coefficient.

In order to cancel out the negative temperature coefficient of the diode voltage, a voltage generator that is proportional to absolute temperature (PTAT) is required. Such a circuit is shown in Figure 3-29. Solving for ΔV_D gives

$$\Delta V_D = V_{D1} - V_{D2} \tag{3.57}$$

$$\Delta V_D = V_t \cdot \ln\left[\frac{nI_0}{I_{S1}}\right] - V_t \cdot \ln\left[\frac{I_0}{I_{S2}}\right] = V_t \ln[n] \tag{3.58}$$

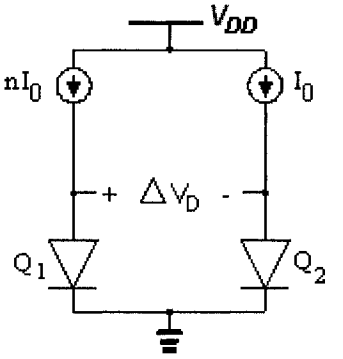


Figure 3-29. A PTAT Circuit

Diode sizes Q_1 and Q_2 are assumed to be the same (i.e. $I_{s1} = I_{s2}$).

The temperature dependence of ΔV_D is thus only k/q, which is a positive quantity. A typical value k/q is 0.087mV/°K. Multiplying this quantity by a factor "c" such that the difference between the temperature coefficient of a diode voltage and "c" times k/q is equal to zero, would result in a temperature independent voltage. Mathematically, subtracting (3.56) from "c" times k/q and solving for "c" gives

$$c = \left[\frac{V_T(4+m) + E_g/q - V_D}{T}\right] \Big/ \frac{k}{q} \tag{3.59}$$

which is approximately 17.2. This means that a multiplication factor of roughly 30 is required, which is too large for practical implementation.

Figure 3-30 shows a schematic of a practical bandgap voltage generator. The output voltage is given as

$$V_{out} = V_D + (V_t \ln[n]) \cdot \left(1 + \frac{R_1}{R_3}\right) \tag{3.60}$$

The error amplifier ensures that V_{D1} is equal to V_{D2} plus the voltage across

resistor R_3. The accuracy of the bandgap voltage depends on inaccuracies in the error amplifier, which include limitations in gain and input offset voltage of the amplifier, and the matching between the two R_1 resistors.

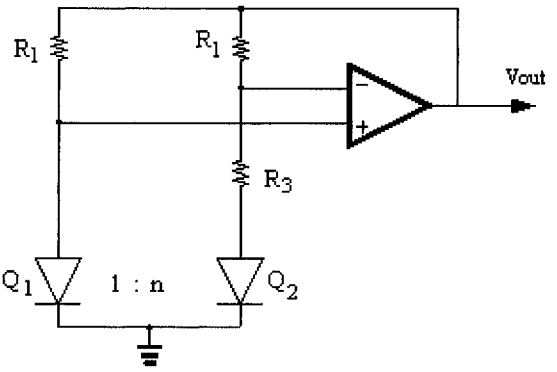


Figure 3-30. Practical CMOS bandgap voltage generator.

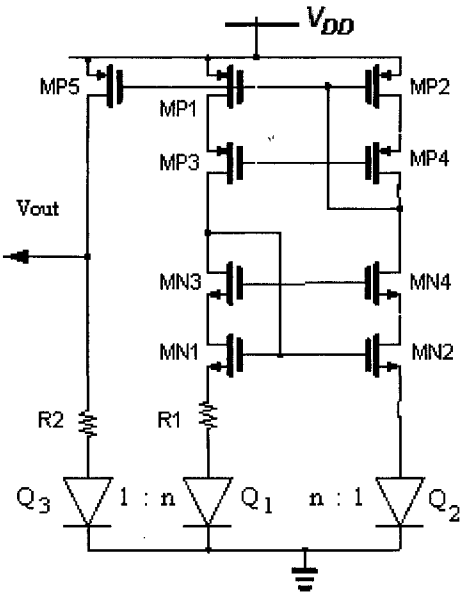


Figure 3-31. Practical PTAT current generator / bandgap voltage generator

Current generators with various temperature behaviours are also possible. Figure 3-31 shows a practical implementation of a PTAT current generator. The core of the PTAT generator consists of transistors MN1-MN4 and MP1-MP4. The current through the two branches are set equal to each other through the cascode feedbacks. Cascode current mirrors are used to improve

matching between the two current branches. One interesting thing to note is that the PTAT current generator can also be used as a bandgap voltage generator by mirroring the current to a resistor and diode as shown in Figure 3-31.

A complementary to absolute temperature (CTAT) current is sometimes desired. Figure 3-32 shows a CTAT current generator. A current proportional to a diode voltage (negative temperature coefficient) is generated by setting a diode voltage equal to the voltage drop across the resistor. As in the case of the PTAT generator of Figure 3-31, the currents through both branches are set equal to each other through cascode feedbacks.

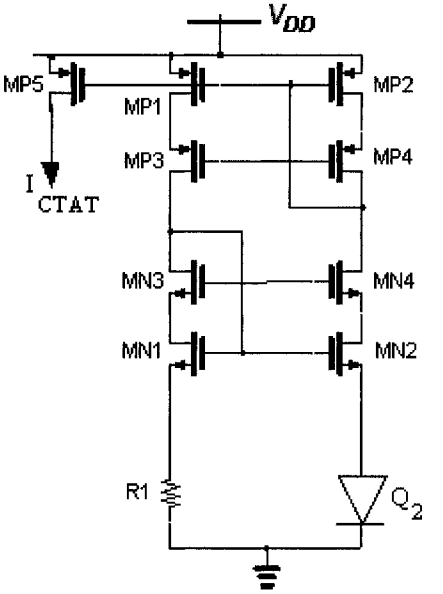


Figure 3-32. Practical CTAT current generator

A bandgap voltage generator can be generated by either combining the currents of CTAT and PTAT generators through mirroring or by altering the bandgap voltage generator of Figure 3-30 to the one shown in Figure 3-33.

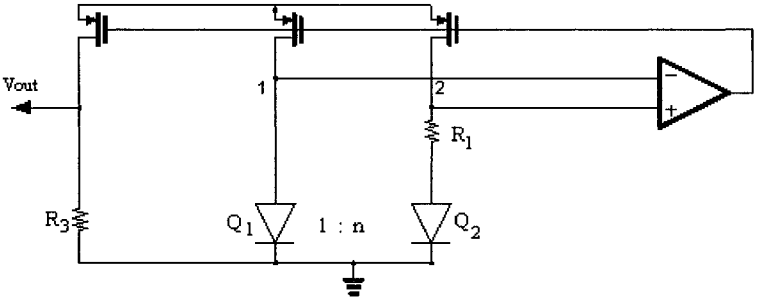


Figure 3-33. Bandgap current generator

For low-voltage operation, it is important to understand the limits on supply voltage scaling for bandgap generators (the same limitations exist in PTAT and CTAT current generators). Assuming the configuration shown in Figure 3-30, there are three primary limitations to the supply voltage. First, the required headroom for the error amplifier is

$$V_{DD,amplifier} = V_D + V_T + 2 \cdot V_{ds,sat} \tag{3.61}$$

where V_D is the diode voltage. A simple differential pair is assumed. Given typical values of 0.7V, 0.5V, and 0.2V for V_D, V_T, and $V_{ds,sat}$ respectively, the minimum possible supply voltage is 1.6V. Another limitation is that the voltage where the diode temperature coefficient cancels the k/q temperature coefficient occurs at roughly 1.2V. To reliably operate a bandgap voltage generator at supply voltage of 1.5V and below, different circuit topologies must be considered.

One such topology is shown in Figure 3-34 [18]. The error amplifier ensures that voltages at nodes 3 and 4 are equal. If R_{1A}=R_{1B} and R_{2A}=R_{2B}, then it follows that the voltages at nodes 1 and 2 are also equal. It also follows that the current in both branches are equal to each other and it can be shown the at the current is equal to

$$I = \frac{V_{D1}}{R_C} + \frac{V_T \ln N}{R_1} \tag{3.62}$$

where R_C=R_{1A}+R_{2A}=R_{1B}+R_{2B}. If all PMOS devices are equal to each other, then the output voltage is equal to

$$V_{out} = \frac{R_3}{R_C}\left[V_{D1} + \left(\frac{R_C}{R_1} \ln N \right) \cdot V_T \right] \tag{3.63}$$

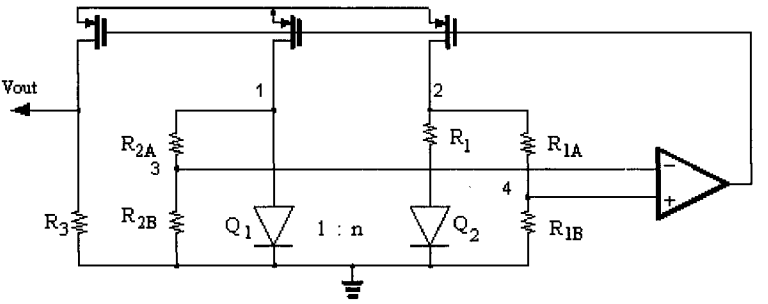


Figure 3-34. A low-voltage bandgap voltage circuit topology

By proper choice of resistors R_c and R_1, a bandgap voltage can be generated. The required voltage headroom in this topology is determined by the minimum allowed voltage input to the error amplifier, which is given as

$$V_{DD,min} = \left(\frac{R_{2A}}{R_{2A} + R_{2B}} \right) \cdot V_D + V_T + 2 \cdot V_{ds,sat} \tag{3.64}$$

by proper choice of resistors, the minimum supply voltage can be lowered by up to 0.7V when compared to conventional techniques.

One question that the reader may ask is, what type of current/voltage generators are best for PLL design? The answer depends on the circuit topologies used in various parts of the PLL. For the VCO, for example, where the speed is slew limited, the delay is proportional to $C_L V_{SW}/I_{SS}$ where C_L is the total parasitic capacitance on a delay cell, V_{SW} is the voltage output swing of the ICO, and I_{SS} is the quiescent current through a delay stage. As the temperature increases, the mobility of the devices is reduced, which reduces the speed of a delay cell. This affects both the center frequency and gain of the ICO. In this case, it is advantageous to have a PTAT current generator for the ICO.

Current must be also generated for the charge pump. The charge pump current determines the gain of the PFD. Assuming that the loop filter components do not vary with temperature, then it is desirable to have a bandgap current for the charge pump. In practice, however, the resistor in the loop filter varies with temperature. If the resistor's temperature coefficient is positive, a CTAT current would help reduce the periodic voltage spikes on the loop filter, called reference spurs (discussed in chapter 4). In order to avoid reducing the loop bandwidth, the V2I current can be made PTAT.

3.7 Summary

In this chapter, some fundamentals of CMOS integrated circuits were reviewed. First, basic device physics of MOSFET devices were presented. This was followed by a more detailed discussion of the circuit implementation of some of the fundamental building blocks of a PLL. One method of enhancing the PLL performance is to increase the maximum allowed voltage headroom on the loop filter, $V_{FILT,max}$. The two key circuit blocks to enable this are the charge pump and the V2I converter. Methods of achieving rail-to-rail operation were discussed. Other key circuit blocks such as VCOs and current and voltage references were also discussed in detail. Although the design of VCOs for functionality may be an easy task, analyzing and optimizing VCOs to achieve a required jitter performance is more challenging, and is the subject of discussion in the next chapter.

REFERENCES

[1] Y. Taur and T. Ning, *Fundamentals of Modern VLSI Devices*, Cambridge: Cambridge Press University, 1998.

[2] W. Liu, *MOSFET Models for SPICE Simulation*, New York: John Wiley & Sons, Inc. 2001.

[3] A. Bellaouar and M. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*, Boston: Kluwer Academic Publishers, 1995.

[4] D. Johns and K. Martin, *Analog Integrated Circuit Design*, New York: John Wiley & Sons, Inc. 1997.

[5] B. Razavi, *Design of Analog CMOS Integrated Circuits*, Boston: McGraw Hill, 2001.

[6] F. Schlogl, H. Dietrich, and H. Zimmermann, "120nm CMOS operational amplifier with high gain down to ±0.3V supply," *IEEE International Systems-on-a-Chip (SOC) Conference*, pp. 121-124, Sept. 2003.

[7] S. Rajput and S. Jamuar, "Low voltage analog circuit design techniques," *IEEE Circuits and Systems Magazine*, vol. 2, issue 1, 2002.

[8] P.K. Chatterjee, et al., "The Impact of Scaling Laws on the Choise of N-Channel or P-Channel for MOS VLSI,"*IEEE Electron Device Letters*, vol. EDL-1, pp. 220-223, October 1980.

[9] L. Sun and D. Nelson, "A 1.0V GHz Range 0.13um CMOS Frequency Synthesizer," *IEEE Custom Integrated Circuits Conference*, pp. 327-330, 2001.

[10] F. You, S. Embabi, and E. Sanchez-Sinencio, "Low-Voltage Class AB Buffers with Quiescent Current Control," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 6, pp. 915-920, June 1998.

[11] Y. Fouzar, Y. Savaria, and M. Sawan, "A CMOS Phase-Locked Loop with an Auto-calibrated VCO," *International Symposium on Circuits and Systems (ISCAS)*, vol. 3, pp. 177-180, 2002.

[12] V. Peluso, et al., "A 900-mV low-power ΔΣ modulator with 12-b dynamic range using the switched-opamp technique," *IEEE J. of Solid-State Circuits*, vol. 32, no. 7, pp. 943-952, July 1997.

[13] W. Wilson, et al., "A CMOS Self-Calibrating Frequency Synthesizer," *IEEE J. of Solid-State Circuits*, vol. 35, no. 10, pp. 1437-1444, Oct. 2000.

[14] J. Maneatis and M. Horowitz, "Precise delay generation using coupled oscillators," *IEEE J. of Solid-State Circuits*, vol. 28, no. 12, pp. 1273-1283, Dec. 1993.

[15] I. Yong, J. Greason, K. Wong, "A PLL Clock Generator with 5 to 100 MHz of Lock Range of Microprocessors," *IEEE J. of Solid-State Circuits*, vol. 27, no. 11, pp. 1599 – 1607, Nov. 1992.

[16] P. Allen and D. Holdberg, *CMOS Analog Circuit Design*, Oxford Press: New York, 1987.

[17] B. Razavi, *RF Microelectronics*, Prentice Hall: Upper Saddle River, NJ, 1998.

[18] N. Leung and P. Mok, "A Sub-1-V 15-ppm/°C CMOS Bandgap Voltage Reference Without Requiring Low Threshold Voltage Device," *IEEE J. of Solid-State Circuits*, vol. 3, no. 4, pp. 526-530, April 2002.

Chapter 4

JITTER ANALYSIS IN PHASE-LOCKED LOOPS

4.1 Introduction

Now that PLL theory and basic analog circuitry have been reviewed, the performance of the PLL is analyzed. The two basic performance metrics of a PLL are jitter and phase noise. Analysis of phase noise in PLLs is well documented and widely available in literature. However, jitter is less understood and in many cases simply stated to be the time-domain equivalent of phase noise. In this chapter, the definitions of jitter are first reviewed. Their relationship to phase noise is more rigorously defined. Jitter in PLLs and all components of PLLs are analyzed. Analysis of sources of jitter and how it affects PLL performance are also analyzed.

4.2 Jitter Basics

One of the most important specifications in microprocessor PLLs is the jitter specification. In short, jitter is a statistical measure of the deviation of the actual PLL clock edges from an ideal clock edges. Non-idealities causing jitter include supply and substrate noise, transistor device noise (mainly thermal and flicker noise), and jitter in the reference signal.

4.2.1 Definition of Jitter

Defining jitter has been a troublesome task since there is no universally accepted terminology for jitter. Jitter definitions include absolute jitter, cycle jitter, cycle-to-cycle jitter, long-term jitter, edge-to-edge jitter, period jitter, and tracking jitter. To make matters worse, different people define

each type of jitter differently. In this section, the types of jitter relevant to microprocessor PLL design is presented and definitions that are accepted by a prevailing majority of literature will be used.

As mentioned earlier, jitter is a *statistical* measure of the deviation of the PLL's actual clock output edges compared to ideal clock edges. This deviation can be measured in a variety of methods. The important thing to note here is that it is a statistical measure of jitter, which means that the worst case jitter for an infinite sample size cannot be determined. For digital applications, it is important that each clock period be longer than a time interval $T_{critical}$ that is needed to execute the critical path under all corner conditions. A suitable definition of jitter in this case would be

$$T_{jitter} = \max_{N} \left\{ \left\| T_{ideal} - T_{min} \right\| \right\} \tag{4.1}$$

where T_{min} is the minimum clock period due to jitter, T_{ideal} is the expected clock period with no jitter, and N is the number of samples collected. T_{jitter} is shown graphically in Figure 4-1. This definition of jitter is called *absolute jitter*. However, since jitter is a statistical quantity, T_{jitter} will only give you the maximum possible jitter with some degree of confidence. The clock period is measured for several samples and subtracted from the ideal clock period. The quantity giving the largest magnitude is given as the absolute jitter. Techniques of choosing a sufficient number of samples to attain a high degree of confidence are covered in Chapter 8. Absolute jitter is sometimes referred to as *time interval error (TIE)* [1].

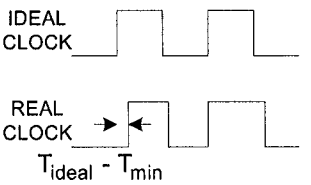


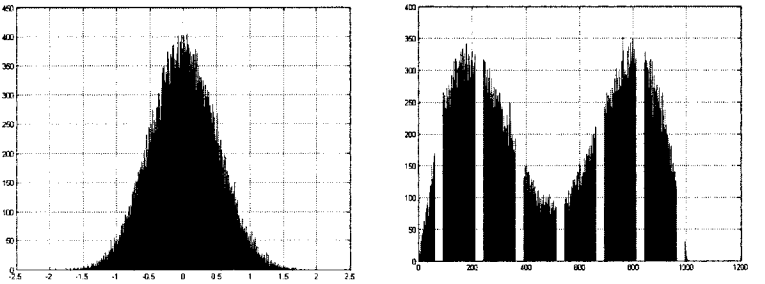
Figure 4-1. Absolute jitter shown graphically

SoC processors usually include data converters that require sampling clocks. One important specification of data converters is the *spurious free dynamic range* (SFDR). The exact definition of SFDR and its impact on the PLL design is covered in Chapter 7. For now, what matters is that data converters are not only sensitive to the absolute jitter, but are sensitive to the shape of the statistical distribution of jitter, especially any kind of tonal jitter given rise to frequency spurs. If the jitter in the output of the PLL follows a Gaussian distribution, then the statistical standard deviation of the jitter, which is given as

$$S_{N,jitter} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(T_i - T_{avg}\right)^2} \tag{4.2}$$

can be used as a measure of jitter, where T_i is the i^{th} sample of the PLL's output period, T_{avg} is the PLL's average period (should be the same as T_{ideal}). The effect of clock jitter on the data converter can then be estimated to the first order by assuming a white noise source with Gaussian distribution.

When the PLL's output exhibits tonal behavior, this can seriously degrade the data converter's SFDR. Using the absolute jitter definition, the histograms of Gaussian distributed jitter and tonal jitter is shown in Figure 4-2. As shown, the histogram of jitter with tonal jitter can have more than one peak (Figure 4-2(b) depicts a bimodal distribution). Also, the shape of the peaks are not Gaussian, they are closer to Raleigh shaped curves. Gaussian distributed jitter is also known as *random jitter* and is typically specified statistically (as an rms value) and is unbounded.



(a) (b)

Figure 4-2. Histogram of jitter with (a) Gaussian distribution and (b) with tonal jitter

The time-domain output of jitter with tonal response resemble very well defined repetitive pattern of jitter as shown in Figure 4-3. At each period, the period error increases by ΔT, until it is reset. This repetitive pattern results in *deterministic jitter*. Deterministic jitter is a bounded type of jitter and is specified as a peak-to-peak jitter. Deterministic jitter can be a dominant component of the total jitter in the PLL's output since it is affected by prevailing factors such as low frequency power supply bounces. Deterministic jitter can also result if both edges of the clock are used and the rise and fall times of the clock are different. Details of measuring deterministic jitter and separating it from random jitter is discussed in more detail in Chapter 8.

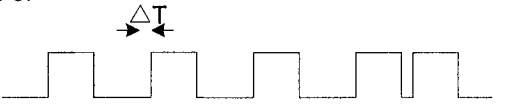


Figure 4-3. Time-domain PLL output with deterministic jitter

Anything that is repetitive in the time-domain shows up as a spur in the frequency domain. In terms of data converter performance, this spur is convoluted with the input signal to the data converter in the frequency domain. Details of this mechanism are covered in Chapter 7.

One way to capture deterministic jitter in the time domain is to measure the change in jitter between two consecutive clock cycles. In other words, measure the *cycle-to-cycle jitter*, which is given as

$$T_{jitter,i} = T_i - T_{i-1} \tag{4.3}$$

where T_i and T_{i-1} are two consecutive clock periods. This can be measured against the maximum cycle-to-cycle jitter, which is also known as *cycle jitter*. If the ratio of cycle-to-cycle jitter to cycle-jitter is large, then that is a good indication that the output contains a large deterministic component. If this ratio is small, then that is a good indication that the output contains a small deterministic component. In some references, cycle-to-cycle jitter is also known as *edge-to-edge jitter* or *adjacent cycle jitter*. Cycle-to-cycle jitter measurement is shown in Figure 4-4.

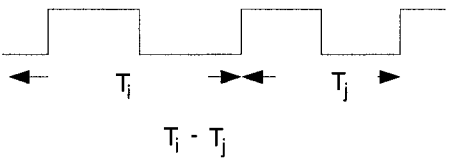


Figure 4-4. Cycle-to-cycle jitter measurement

Other definitions of jitter are also used in the literature. For instance, *period jitter* is sometimes used and is defined as the variation in clock period due to jitter. This is similar to cycle jitter, except that the variation is measured across all clock period samples. It effectively portrays the same type of information as absolute jitter, except that it can be described as a more general statistical measure, rather than a worst case statistical measure of jitter.

4.2.2 Sources of Jitter

Now that we are familiar with the definitions of jitters, the next question should be what are the sources that give rise to jitter? As alluded to earlier, there may be intrinsic and extrinsic sources of jitter. Intrinsic sources of jitter arise from the voltage and current noise sources of the devices used in the various PLL components. Intrinsic sources of jitter also include any loop instabilities that cause undesired oscillations, or jitter, at the output. This can be due to an underdamped or marginally stable PLL response. Such noise

sources undergo an amplitude-to-phase noise conversion with transfer functions that depend on the noise injection point in the PLL. Extrinsic noise sources include supply and substrate bounces, and noise from the crystal reference. Although the crystal reference is normally a very low jitter reference signal, significant amount of jitter can be added before it reaches the PLL since the crystal is usually off-chip. Other sources of external jitter can include crosstalk (whether magnetic or electrical) from nearby high-speed digital lines coupling into sensitive PLL nodes (such as the loop filter node). For very high clock rates (>1GHz), on-chip transmission losses and reflections that alter the zero crossings of the clock signal may cause jitter. External noise sources also undergo an amplitude-to-phase noise conversion process with transfer functions that depend on the noise injection point in the PLL.

4.2.3 Jitter and Phase Noise

First, the distinction between phase noise and amplitude noise is made clear. A non-ideal sine wave may be given as

$$V(t) = V[1+n_1(t)]\sin[\omega_t t+n_2(t)] \tag{4.4}$$

Consider the sine wave shown in Figure 4-5(a). Due to any noise source injected into the sine wave, the sine wave may be corrupted by both additive noise and phase noise as shown in Figure 4-5(b) and 4-5(c), respectively. In the case of additive noise, only the amplitude is affected (i.e. $n_1(t)$ represents additive noise in the above equation); whereas in the case of jitter, horizontal excursions superimposed on the sine wave exist (i.e. mathematically, this is represented by $n_2(t)$ in the above equation). This results in an error term associated with each period of the synthesized sine wave. For example, if a period P=5ns is required, at a certain instance a period of 5.1ns may be produces; whereas in another instance a period of 4.8ns may be produced.

Now, a first order mathematical analysis of jitter is given. Consider a sine wave corrupted by jitter given as

$$V(t) = V[1+\sin[\omega_t t+n_2(t)] \tag{4.5}$$

where $n_2(t)$, in general, is some random variable representing jitter. To simplify the analysis let $n_2(t)$ by a sine function. Therefore, (4.5) becomes

$$V(t) = V[1+\sin[\omega_t t+\Delta f/f_m \sin\omega_m t]] \tag{4.6}$$

where Δf is the peak frequency deviation and $\theta_p=\Delta f / f_m$ is the peak phase deviation, which is also known as the modulation index. This shows that jitter results in FM modulation of a carrier frequency with the frequency spectrum of the jitter. Expanding (4.6) yields

$$V(t) = V + V\{\cos(\omega_c t)\cdot\cos(\theta_p \sin(\omega_c t)) - \sin(\omega_c t)\sin(\theta_p \sin(\omega_c t))\} \quad (4.7)$$

Using the Bessel function [2], V(t) can be expressed as

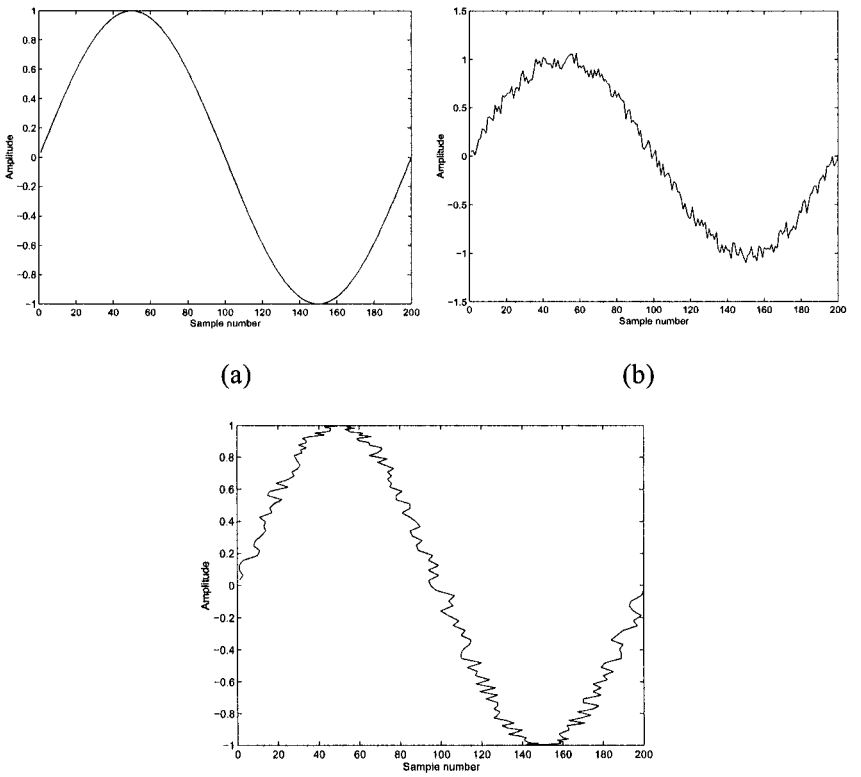


Figure 4-5. Sine wave in (a) ideal form, (b) corrupted by additive noise, (c) or corrupted by phase noise

$$V(t) = V + V\{\{\cos(\omega_t t)\{J_0(\theta_p) + 2\sum_{k=2i} J_k(\theta_p)\cos(k\omega_t t)\} - \{2\sum_{k=2i} J_k(\theta_p)\sin(k\omega_t t)\}\} \quad (4.8)$$

If it is assumed that $\theta_p << 1$ (i.e. small phase noise),

$$\cos(\theta_p \sin(\omega_t t)) \approx 1 \tag{4.9}$$

and

$$\sin(\theta_p \sin(\omega_t t)) \approx \theta_p \sin(\omega_t t) \tag{4.10}$$

therefore, (4.7) can be simplified to

$$V(t) = V[\cos(\omega_t t) - \sin(\omega_t t)(\theta_p \sin(\omega_t t))]$$

$$= V\{\cos(\omega_t t) - (\theta_p/2)[\cos(\omega_t+\omega_m)t - \cos(\omega_t - \omega_m)t]\}. \tag{4.11}$$

This means that as long as the frequency (or phase) deviation of the clock jitter is small, the output spectrum yields a single pulse at the desired frequency and two sidebands at frequencies $\omega_t \pm \omega_m$ and their amplitude is equal to $\frac{1}{2} \cdot V \cdot \theta_p$.

The single sideband phase noise is given as

$$L(f_m) = (V_{noise}/V_{signal})^2 = \theta_p^2/4 = \theta_{rms}^2/2 \tag{4.12}$$

Since it was assumed that $\theta_p << 1$, the double sideband phase noise is given as

$$S_\varphi(f) = 2\, L(f_m) = \theta_p^2/2 = \theta_{rms}^2 \tag{4.13}$$

It is important to remember that when measuring these quantities, θ_p and θ_{rms} are both measured over a 1-Hz bandwidth. In general, the frequency spectrum of the jitter will contain many spectral components. Summing the power spectral density (psd) of the phase noise over a frequency band of interest results in the rms jitter. This assumes that the psd of the phase noise is a stationary statistical quantity. Deterministic jitter is incorporated by summing the individual jitter contribution of each spurious signal to the overall jitter. In general, obtaining rms jitter from a given phase noise spectrum is a straight forward matter. However, the reverse more difficult since much of the spectral information is lost in a jitter measurement.

4.3 Jitter in Voltage Controlled Oscillators

Voltage controlled oscillators (VCOs) usually contribute most to the total jitter of a PLL. In this section, jitter in VCOs is analyzed in detail. First,

classical analysis of phase noise is given. The more general case of time-varying, or cyclostationary, random noise is then analyzed. Common CMOS implementations of both ring oscillator and LC VCOs are analyzed and jitter expressions are derived. The output of the voltage-controlled oscillator (VCO) is a large swing waveform, and thus small signal AC analysis does not fully describe its operation.

4.3.1 Classical Phase Noise Analysis

In its most general form, an oscillator may be represented by the positive feedback system shown in Figure 4-6. The transfer function of this system is

$$H(j\omega) = \frac{F(j\omega)}{1 - F(j\omega)} \tag{4.14}$$

where $F(j\omega)$ is the open loop transfer function of the oscillator, and $H(j\omega)$ is the closed loop transfer function of the oscillator. Since it is a positive feedback system, the amplitude of oscillation will grow until a nonlinearity in the oscillator causes its amplitude to saturate. Such nonlinearities include power supply limitations or transistor saturation. At large amplitudes, circuit nonlinearities become so severe that the gain of $F(j\omega)$ becomes 1 and the total phase shift around the feedback loop is $0° \pm 360 \cdot n°$, where n is a positive integer. These two requirements constitute what is known as the *Barkhausen's* criteria for oscillation. When $F(j\omega_o)=1$, the oscillator's frequency is ω_o rad/s.

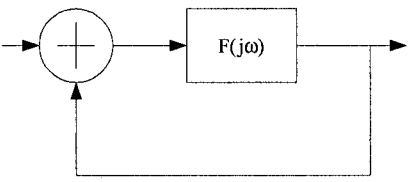


Figure 4-6. A generic positive feedback system

Another way of looking at a system representing an oscillator is by looking at the root locus of the system in Figure 4-6 while varying $F(j\omega)$. Initially $F(j\omega)$ is much greater than one. As the voltage swing grows, the magnitude of $F(j\omega)$ quickly decreases up until the poles of the system are on the imaginary axis. At this point, stable oscillation is sustained at a certain frequency ω_o. At this point, the steady-state magnitude of the system is exactly one. Any perturbation in the magnitude would cause the poles to shift to the right or left causing a frequency or phase shift. This translation of magnitude error to frequency or phase error is what causes jitter. The root locus plot of the system in Figure 4-6 is shown in Figure 4-7.

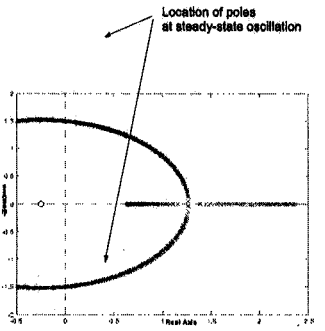


Figure 4-7. Root locus of positive feedback system

An important parameter of an oscillator's performance is its *open loop Q factor*. Simply stated, the open loop Q factor is a measure of how much the closed loop feedback system opposes variation in frequency of oscillation. One commonly used equation for open loop Q factor is

$$Q = \frac{\omega_o}{\Delta\omega} \tag{4.15}$$

where ω_o is the oscillation frequency and $\Delta\omega$ is the double sided frequency offset where the magnitude of the transfer function is one-half its peak value at ω_o as shown in Figure 4-8. Using this definition, the magnitude of $H(j\omega)$ at $\omega_o+\Delta\omega$ is

$$|H(j\omega)| = \frac{R_p}{2} = \frac{R_p}{2}\left(\frac{\omega_o}{Q \cdot \Delta\omega}\right) \tag{4.16}$$

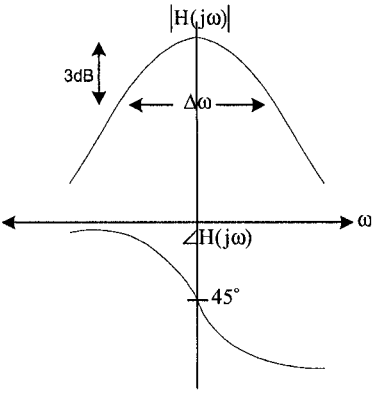


Figure 4-8. Bode plot of closed loop transfer function of oscillator

Assuming that the dominant component of noise is thermal noise, the power spectral density of the thermal noise shaped by the oscillator transfer function is

$$\frac{\overline{v_n^2}}{\Delta f} = \frac{\overline{i_n^2}}{\Delta f} \cdot |H(j\omega)|^2 = 4kTR_p \cdot \left(\frac{\omega_o}{2Q\Delta\omega}\right)^2 \tag{4.17}$$

Equation (4.17) gives the total output spectrum including the amplitude as well as phase spectrum. Using the equipartition theorem of thermodynamics [3], the total spectrum is split evenly between phase noise spectrum and amplitude noise spectrum. However, since the oscillator nonlinearities provide an indirect form of amplitude control, the total output spectrum of the oscillator is given as only half of equation (4.17). Also, phase noise is normally reported as relative to the carrier signal at ω_o. Using these two facts, the phase noise spectrum of an oscillator dominated by thermal noise is given as

$$L\{\Delta\omega\} = \frac{2kT}{P_c} \cdot \left(\frac{\omega_o}{2Q\Delta\omega}\right)^2 \tag{4.18}$$

where P_c is the amplitude power of the oscillator. This formula is known as *Leeson's* formula [4]. Given this expression, the rms phase error can be found by summing equation (4.18) over the entire noise bandwidth. Since much of the power of the phase noise lies within a frequency bandwidth of $\Delta\omega$ around ω_o, the rms phase error can be found as

$$\theta_{e,rms} = \int_{-\frac{\omega_o}{2Q}}^{\frac{\omega_o}{2Q}} \frac{2kT}{P_c} \cdot \left(\frac{\omega_o}{2Q\Delta\omega}\right)^2 d(\Delta\omega) \tag{4.19}$$

Evaluating this integral yields the following result:

$$\theta_{e,rms} = \frac{kT}{P_c} \cdot \left(\frac{\omega_o}{Q}\right) \text{ rads} \tag{4.20}$$

Converting radians to time, the rms jitter $T_{j,rms}$ is given as

$$T_{j,rms} = \frac{kT}{2P_c} \cdot \left(\frac{1}{Q}\right) \text{ sec} \tag{4.21}$$

This is a very significant result. It clearly shows that for a given VCO topology (fixed Q), the rms jitter varies only with the power level of the oscillator output signal. *This means that there is a direct trade-off between power consumption and jitter in oscillators.*

The above discussion assumes that the noise of the oscillator depends only on the thermal noise of the devices used to build the oscillator. Such an assumption leads to the conclusion that the phase noise decreases at a rate of 20dB/decade indefinitely. Empirical data shows that this is generally not true. A more accurate plot of phase noise spectrum of an oscillator is shown in Figure 4-9.

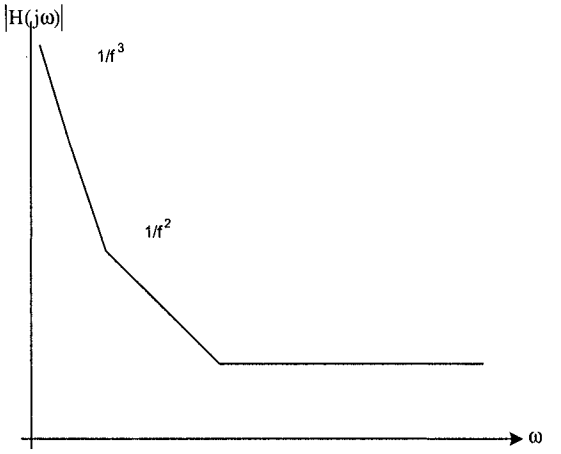


Figure 4-9. More realistic phase noise spectrum of an oscillator

An equation representing the graph in Figure 4-9 can be give as

$$L\{\Delta\omega\} = \frac{2kTF}{P_c} \cdot \left[1 + \left(\frac{\omega_o}{2Q\Delta\omega}\right)^2\right] \cdot \left[1 + \frac{\Delta\omega_{1/f}^3}{|\Delta\omega|}\right] \tag{4.22}$$

where F is a empirical parameter (or "fudge" factor) and $\Delta\omega_{1/f}^3$ is the corner frequency between the $1/f^2$ and $1/f^3$ regions of the phase noise spectrum. Equation (4.22) is known as the *Leeson-Cutler* formula [5]. The F parameter accounts for other noise sources other than the losses in the F(jω) system. Such noise sources can include device noise and supply noise. The "1" factor accounts for the fact that there is a phase noise floor that defines the absolute minimum phase noise achievable at all offset frequencies. The $1/f^3$ region is caused by upconversion of 1/f noise (mainly device flicker noise) to near the oscillation frequency. The Leeson-Cutler formula suggests that the boundary between the $1/f^3$ and $1/f^2$ regions is exactly the boundary

between the 1/f and thermal noise regions. However, empirical data suggests otherwise for reasons that are explained in the next section.

4.3.2 Cyclostationary Phase Noise Analysis

Figure 4-10 shows a typical VCO waveform with noise injected into it. As the figure shows, the voltage swing can be a significant fraction of the supply voltage. Under such large voltage swings, small signal analysis breaks down. This is because transistors that make up the VCO can enter different regions of operation (saturation, linear, or cut-off). The jitter performance of the VCO has to be analyzed at different time instances where the VCO devices operate in the various regions.

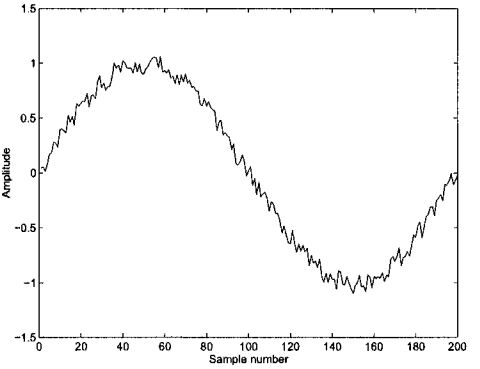


Figure 4-10. Typical VCO waveform with noise injection

Figure 4-11 shows the translation of voltage noise into phase error at various regions of operation. As the figure shows, the amount of jitter resulting from the same amplitude can differ drastically depending on when the noise sample is injected into the VCO. For this reason, the expression for jitter in VCOs is a time-varying function. Moreover, since the output of a VCO is periodic, the noise sources that induce jitter can periodically with time. Such noise sources are called *cyclostationary* noise sources [6].

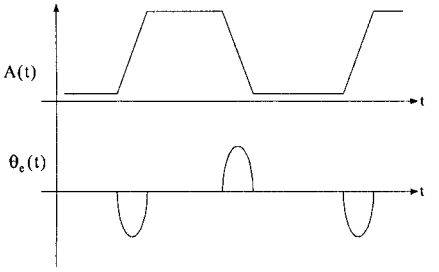


Figure 4-11. Jitter transfer function for various noise injection times

The output of an oscillator may be given as

$$V_{out}(t) = A(t) \cdot f[\omega_0 t + \phi(t)] \tag{4.23}$$

where the function f is periodic in 2π and $\phi(t)$ and $A(t)$ are the phase and amplitude variations due to noise, respectively. Amplitude variations may alter the zero-crossing of the oscillator's output, and hence translate into phase variation. If the amplitude variation is small enough, the amplitude-to-phase translation may be assumed to be linear. An instantaneous change in voltage would cause an instantaneous change in charge, which is given by

$$\Delta V = \frac{\Delta q}{C_{node}} \tag{4.24}$$

where C_{node} is the capacitance of the node which experienced charge injection due to noise. The phase variation can be given by [18]

$$\Delta\phi = \Gamma(\omega_0 t)\frac{\Delta q}{q_{swing}} \tag{4.25}$$

where $q_{swing} = C_{node} \cdot V_{swing}$, and V_{swing} is the voltage swing of the node which experienced the noise charge injection. $\Gamma(\omega_0 t)$ is a unitless time-varying function, called the *impulse sensitivity function (ISF)* [7]. When an internal signal is high or low, amplitude noise will have little or no effect on phase error on the output of the ring oscillator. This means that the ISF is a small or zero value during that interval. On the other hand, the ISF is maximized at the time of a transition switching of an internal node. Note that once a phase error has occurred, it is not corrected for. Therefore, the phase error accumulates indefinitely as time increases. This type of analysis assumes that the oscillator is a linear time-varying (LTV) system.

Assuming that the total noise in the oscillator can be represented as current noise i(t), the output phase error of the oscillator is given as

$$\phi(t) = \frac{1}{q_{max}} \int_{-\infty}^{t} \Gamma(\omega_0 \tau) i(\tau) d\tau \tag{4.26}$$

where q_{max} is the maximum charge injected by the noise source. Since ISF is a periodic function, it can be represented as a Fourier series

$$\Gamma(\omega_0 \tau) = \frac{c_0}{2} + \sum_{1}^{\infty} c_n \cos(n\omega_0 \tau) \tag{4.27}$$

Substituting (4.27) back into (4.26) gives

$$\phi(t) = \frac{1}{q_{max}}\left[\frac{c_0}{2}\int_{-\infty}^{t} i(\tau)d\tau + \sum_{1}^{\infty} c_n \int_{-\infty}^{t} i(\tau)\cos(n\omega_o\tau)d\tau\right] \tag{4.28}$$

Consider two cases for i(t): (a) when it is a low frequency sinusoidal signal with frequency $\Delta\omega$, and (b) when it is a sinusoidal signal with a frequency near the carrier ω_o with frequency $\omega_o \pm \Delta\omega$. In the first case, only the first integral in (4.25) contains a significant signal, and the resulting output phase error is given as

$$\phi(t) = \frac{I_0 c_0 \sin(\Delta\omega t)}{2q_{max}\Delta\omega} \tag{4.29}$$

where I_0 is the maximum amplitude of the input current. A similar result can be shown for case (b), but with Fourier coefficient c_1. More generally, it can be shown that for a current sinusoidal input with frequency $\omega_o \pm n\Delta\omega$, the output phase error will be a sinusoid with frequency $\Delta\omega$ and magnitude proportional to c_n. These phase errors are then upconverted to ω_o (for c_0) and downconverted to ω_o (for c_2, c_3, ..., c_n) to the oscillation frequency by the nonlinear oscillator transfer function. Noise sources near ω_o remain the same. This means that noise injected into the oscillator is only significant if it is near dc, near the oscillator frequency, or a harmonic of the oscillator frequency. A similar result was reached in [27].

The statistics of the timing jitter depends on the correlation of the noise sources involved. In the case of thermal noise, the noise sources are considered to be random and uncorrelated. Therefore, it follows that for a ring oscillator with identical stages, the total variance of the jitter will be given as the sum of the variance of the individual sources of noise. This means that for a given observation time interval ΔT, the standard deviation of the jitter of the ring oscillator due to thermal noise is given as [16]

$$\sigma_{\Delta T} = \kappa\sqrt{\Delta T} \tag{4.30}$$

where κ is a proportionality constant which can be shown to be equal to

$$\kappa = \frac{\Gamma_{rms}}{q_{max}\omega_0}\sqrt{\frac{1}{2}\frac{\overline{i_n^2}}{\Delta f}} \tag{4.31}$$

where ω_0 is the output target frequency in rads/sec, q_{max} is equal to $C_L \cdot V_{SW}$, where C_L is the parasitic capacitance at the output of the oscillator and V_{SW} is the voltage swing of the oscillator, and $\frac{\overline{i_n^2}}{\Delta f}$ is the power spectral density of

the thermal noise of the active devices in the oscillator. For CMOS transistors, the drain current noise spectral density is given by [31]

$$\frac{\overline{i_n^2}}{\Delta f} = 4kT\gamma\mu C_{ox}\frac{W}{L}(V_{GS} - V_T)^2 \tag{4.32}$$

where γ is a coefficient equal to 2/3 for long-channel transistors. As equations (4.31) and (4.32) show, jitter due to thermal noise can be minimized by reducing the ISF, increasing the voltage swing (at the expense of power consumption), increasing the operating frequency, and reducing the power spectral density of the device thermal noise.

Correlated noise sources are usually a result of low frequency noise, such as flicker noise, as well as power supply bounces. Flicker noise can be minimized by using large transistors. One interesting result from ISF analysis is that the corner frequency of $1/f^2$ and $1/f^3$ can be accurately determined to be [19]

$$\omega_{1/f^3} = \omega_{1/f}\cdot\left(\frac{\Gamma_{dc}}{\Gamma_{rms}}\right)^2 \approx \omega_{1/f}\cdot\left(\frac{c_0}{c_1}\right)^2 \tag{4.33}$$

where $\omega_{1/f}$ is the corner frequency of 1/f noise and thermal white noise. Therefore, the upconversion of flicker noise can be minimized by having a more symmetric waveform around the x-axis (for zero dc level) and by maximizing the oscillator's voltage swing.

The effect of power supply bounces can be minimized by employing circuit techniques which have good positive and negative PSRR. Note that PSRR is not a measure of jitter; it is merely a measure of how much variation in the power supplies translates into variation on the input/output nodes of the oscillator. The translation of the noise on the input/output of the oscillator to phase variation is described by the ISF. The effect of power supply and ground bounces on the oscillator's performance is analyzed in more detail in Section 4.5.

4.3.3 CMOS Voltage Controlled Ring Oscillators

The most popular configuration for a VCO for microprocessor PLLs is the ring oscillator configuration. A ring oscillator consists of N inverting or buffer stagers with the last stage feeding the first such that a total phase shift of 360 degrees is achieved. To reduce power consumption, a three stage ring oscillator is usually desired [23-25]. In this section, two popular inverting stages are considered and the performance of oscillators using the two different types of inverting stages are evaluated. The ISF as well as both the

positive and negative power supply rejection ratio (PSRR) are used as figure of merits for the oscillator's performance.

The simplest realization of a delay stage in a ring oscillator is a CMOS inverter. Figure 4-12 shows a three stage CMOS inverter-based ring oscillator. When steady-state oscillation is reached, the phase shift through each stage is 60 degrees. A 180 degrees phase shift is achieved by having an inverting feedback signal. This gives a total of 360 degrees of phase shift, which is required for oscillation.

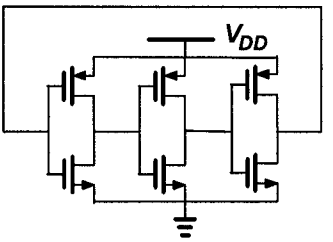


Figure 4-12. A CMOS inverter-based ring oscillator

A plot of the ISF for one period of oscillation is shown in Figure 4-13. The average value of the ISF is 0.064 and the rms value is 1.291. Figure 4-14 shows both the positive and negative DC power supply rejection ratio (PSRR) of a single stage of the oscillator versus the input voltage swing. As shown in the figure, the oscillator has poor PSRR+ at low input voltages (since the PFET is turned on, and thus allowing noise on VDD to couple onto the output node). The figure also shows that the oscillator has poor PSRR- at high input voltages (since the NFET is turned on, and thus allowing noise on VSS to couple onto the output node).

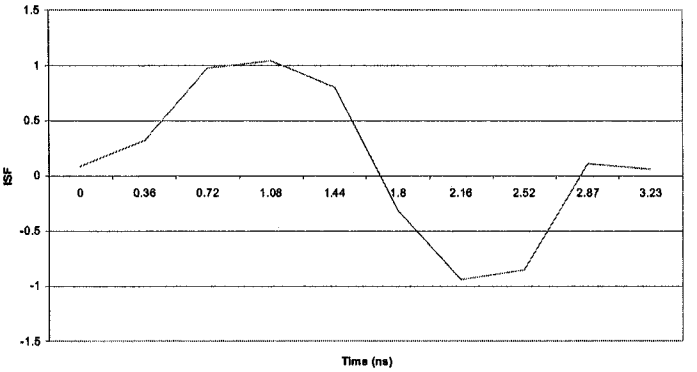
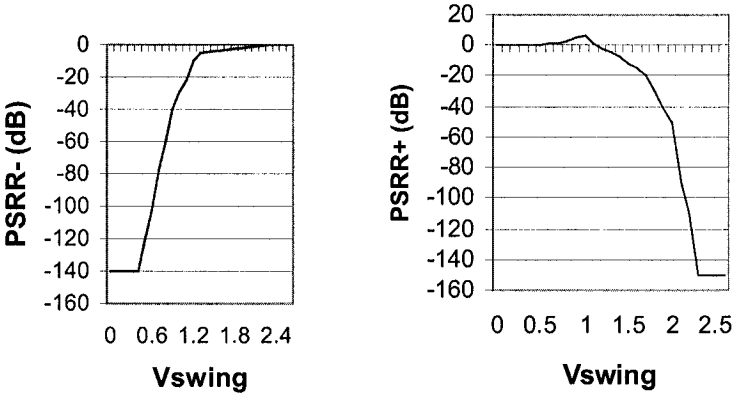


Figure 4-13. ISF for CMOS inverter-based ring oscillator

Next, a differential pair with Maneatis loads [8] is evaluated as shown in Figure 4-15. The ring oscillator is composed of three of these delay cells along with a replica bias circuit. The replica bias circuit is necessary to provide a bias voltage for the top PMOS load devices. Diode PMOS load devices are also used to limit the swing to a V_T drop. The NFET devices provide the bias current. This bias current is cascoded to prevent variation of current with voltage biasing. Varying the current results in variation in frequency. Figure 4-16 shows the ISF of the oscillator. The DC average of the ISF is 0.028 and its rms value is 1.10. The negative PSRR was limited to that of a cascode mirror, but degrades at high frequency. PSRR+ was even better than PSRR- since both the input and output are referenced to VDD. The PSRR is shown in Figure 4-17.



(a) (b)

Figure 4-14. PSRR for CMOS inverter-based ring oscillator: (a) PSRR- and (b) PSRR+

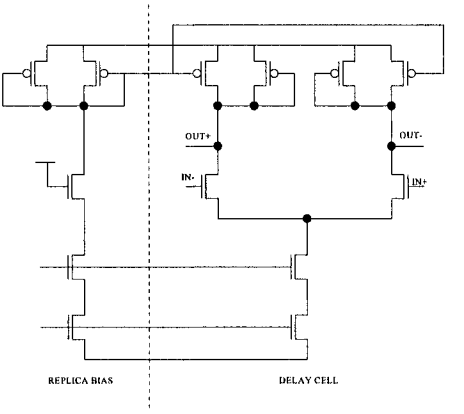


Figure 4-15. Ring oscillator with PMOS Maneatis loads

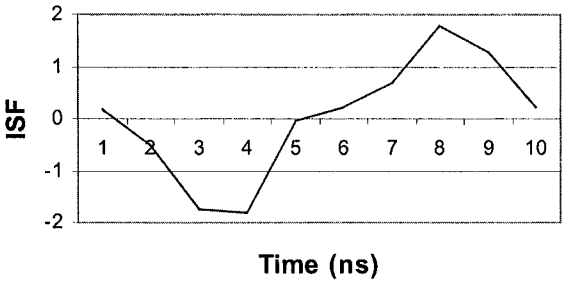


Figure 4-16. ISF of ring oscillator with PMOS Maneatis loads

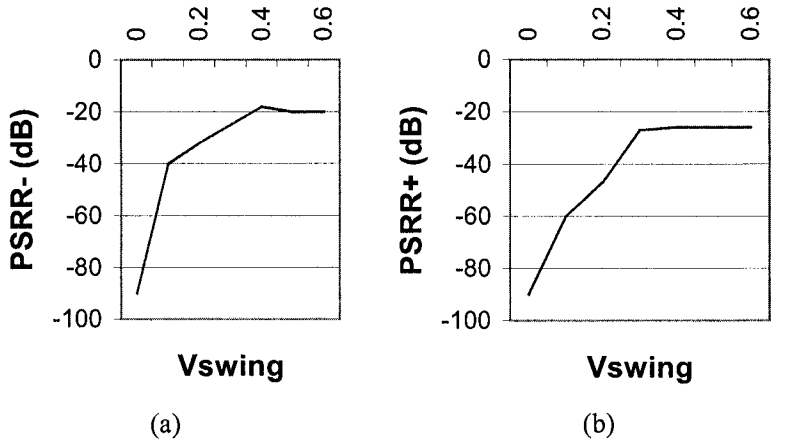


Figure 4-17. PSRR of ring oscillator with PMOS Maneatis loads

Table 4-1 summarizes the performance of the various VCO topologies studied in the previous section. The table evaluates each VCO based on the range of both positive and negative PSRR, as well as the rms and average values of the ISF, and the ratio of the average value of ISF to the rms value of ISF. This ratio determines how much correlated noise (1/f and power supply bounces) translates into jitter. As the table reveals, the differential VCO topologies have superior PSRR+/- to the single-ended topologies. In terms of both PSRR and $\frac{\Gamma_{dc}}{\Gamma_{rms}}$ ratio, the best choice is the differential pair with NMOS loads.

Table 4-1: Summary of VCO Performance

VCO Type	Γ_{rms}	Γ_{dc}	$\frac{\Gamma_{dc}}{\Gamma_{rms}}$	PSRR+	PSRR-
CMOS inverters	1.31	0.059	0.0450	0dB	0dB
Diff pair (PMOS loads)	1.12	0.032	0.0286	-29.3dBc	-21.5dBc

4.3.4 Jitter in Differential LC VCOs

Another popular VCO topology is the LC VCO. The LC VCO operates by resonating a high-Q tank at the desired output frequency. A negative g_m stage sustains the oscillation by compensating for the losses in the LC tank, as explained in the previous chapter. LC VCOs are often used in wireless systems where much more stringent phase noise specifications must be met. The cost and difficulty of integrating the inductor in an LC VCO makes it unattractive for implementation in microprocessor PLLs. Another disadvantage of LC VCOs is that their tuning ranges are often much narrower than ring oscillators [9]. Figure 4-18 shows a typical LC VCO implemented in CMOS technology.

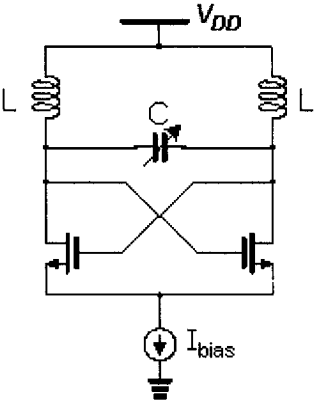


Figure 4-18. CMOS LC VCO

Nonetheless, due to more stringent jitter requirements in SoC processors, some microprocessors contain LC-based VCOs in the clock generation PLL designs [32]. An equivalent model of the LC VCO is shown in Figure 4-19, where the losses in the tank are represented by g_{tank} and the negative-g_m due to the cross-coupled FETs is represented by $-g_{FET}$. In order to sustain

oscillation, the small signal loop gain must be greater than one. This is achieved by canceling the tank losses, g_{tank}, by ensuring that $g_{FET} > g_{tank}$. The oscillation amplitudes will continue to increase until a nonlinearity is reached that causes the large signal gain to be one at the desired oscillation frequency. This nonlinearity may be supply voltage headroom or bias current limitation across the real impedance of the tank. When the real component in the tank impedance is canceled out, the oscillation frequency is given as

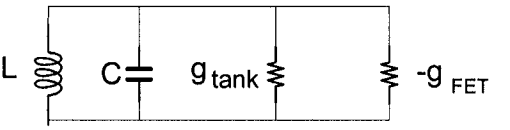


Figure 4-19. Equivalent model of cross-coupled FET LC oscillator

$$\omega = \sqrt{\frac{1}{LC}} \tag{4.34}$$

where L is the effective tank inductance at resonance frequency ω and C is the total capacitance at the tank's node (including parasitic capacitance). The amplitude of the tank is

$$V_{tank} = \frac{I_{bias}}{g_{tank}} \tag{4.35}$$

As long as the amplitude of oscillation is not limited by the circuit's voltage headroom, increasing the energy in the tank will lead to reduction in phase noise and jitter. Energy stored in the LC tank is

$$E_{tank} = \tfrac{1}{2} C V_{tank}^2 \tag{4.36}$$

Substituting (4.35) and (4.34) into (4.36) and solving for the tank voltage gives

$$V_{tank}^2 = 2E_{tank}\omega^2 L \tag{4.37}$$

Intuitively, this equation would lead one to believe that increasing the inductance alone would result in lower phase noise. However, in an LC oscillator, the tank's voltage and thermal noise voltage increase at the same rate. Under the simplifying assumption that the noise is dominated by upconverted thermal noise ($1/f^2$ noise), the ratio of the tank voltage to thermal noise voltage can be shows to be [33]

$$\frac{V_{tank}^2}{\left\langle v_n^2 \right\rangle} = \frac{2E_{tank}}{kT} \tag{4.38}$$

where k is Boltzmann's constant. Solving (4.37) for tank energy, using (4.35), and substituting into (4.38) gives

$$\frac{V_{tank}^2}{\left\langle v_n^2 \right\rangle} = \frac{I_{bias}^2}{\omega^2 \left(g_{tank}^2 L\right) kT} \tag{4.39}$$

This equation shows that in order to maximize the amplitude to noise ratio, the product of $g_{tank}^2 L$ must be minimized. Increasing the bias current also helps to improve the noise performance. This equation assumes that the amplitude of oscillation is not limited by supply voltage.

The tank's inductance and parallel conductance are, however, related. To understand this, a widely used inductor model is reviewed [34]. Figure 4-20 shows a relative complete on-chip spiral inductor model.

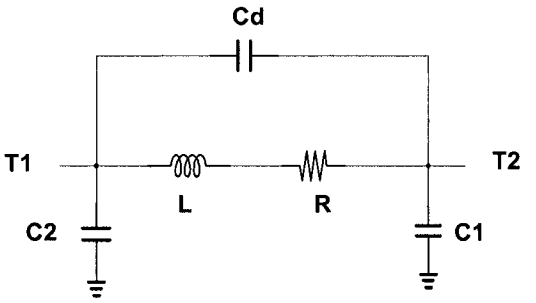


Figure 4-20. Spiral inductor model

T1 and T2 are the two terminals of the spiral inductor. Cd is the distributed capacitance of the inductor, C_1 and C_2 are the fringing capacitance, and L is the effective inductance, and R is the series resistance of the spiral inductor. The tank inductance (in Henries) is given as

$$L \approx \frac{37.5\mu_0 n^2 a^2}{22r - 14a} \tag{4.40}$$

where a is the mean radius of the inductor, r is the outermost radius, n is the number of turns of the spiral inductor, and $\lambda\mu_0$ is the permeability of free space, which is $4 \cdot \pi \times 10^{-7}$ T^2m^3/J. The series resistance (in ohms) is given as

$$R = \frac{\ell}{w \cdot \sigma \cdot \delta \cdot \left(1 - e^{-t/\delta}\right)} \tag{4.41}$$

where w is the width, ℓ is the total length, σ is the conductivity, and δ is the skin depth of the metal spiral inductor. The effective conductance of the tank is inversely related to the metal series resistance. Assuming the effect of skin depth is small and the width of the metal is much larger than the spacing between metal turns, it can be shown that tank parallel conductance, g_{tank}, is inversely proportional to the square root of the area of the inductor. More importantly, the $g_{tank}^2 L$ product is proportional to the square root of the area. This means that for constant area, the effective quality of the inductor can only be enhanced by having a smaller inductor. If the area is not constrained, the quality of the tank can be enhanced by using larger metal widths. The metal width can be increased until the tank's quality starts to degrade due to excessive capacitive losses through the substrate.

4.4 Jitter Performance of Closed-Loop PLL System

In this section, the closed-loop PLL jitter performance is evaluated. First, basic linear analysis of the PLL is reviewed. The contribution of the VCO's jitter to the overall PLL jitter is analyzed. This is followed by a more in-depth analysis of varying loop parameters on jitter. Such parameters include phase detector gain and VCO gain. Non-linear effects such as PFD delay and charge pump imbalances are reviewed.

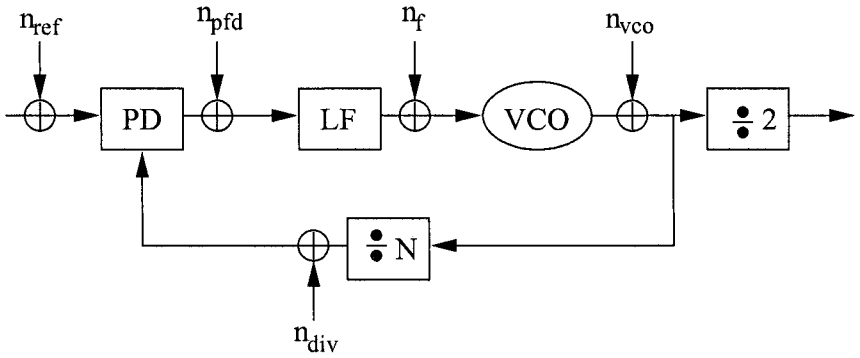


Figure 4-21. Linear model of PLL

4.4.1 Basic Linear Analysis

Assuming the sources of jitter are independent and the PLL is linear, the total jitter is the sum of the individual jitter contribution from all the noise sources. Noise can be coupling from either the substrate or the power supply line. The amplitude of coupling of noise from power supply is usually much greater than that of the substrate. Thermal noise is usually white

uncorrelated noise. The transfer function of the closed loop PLL from each node in Figure 4-21 to the output can be expressed as

$$H_{ref}(s) = \frac{K_d K_o F(s)/s}{1 + K_d K_0 F(s)/Ns} \tag{4.42}$$

$$H_{pfd}(s) = \frac{K_o F(s)/s}{1 + K_d K_o F(s)/Ns} \tag{4.43}$$

$$H_f(s) = \frac{K_o/s}{1 + K_d K_o F(s)/Ns} \tag{4.44}$$

$$H_{div}(s) = \frac{K_d K_o F(s)/s}{1 + K_d K_o F(s)/Ns} \tag{4.45}$$

$$H_{vco}(s) = \frac{1}{1 + K_d K_o F(s)/Ns} \tag{4.46}$$

Note that only Equation 4.46, corresponding to the VCO input, is a high pass filter. All the rest are low pass and band pass filters. Therefore, the noise from PLL components other than the VCO is attenuated beyond the PLL's closed loop bandwidth. The term "F(s)" denotes the loop filter transfer function. Closed loop PLL expressions assuming first order and second order loop filter transfer functions are shown in Equations (2.9) and (2.17), respectively.

4.4.2 Effect of Varying Loop Parameters

In analyzing the effect of loop parameters on PLL jitter, a method of quantifying the jitter shaped by the PLL must be established. A commonly used measure of jitter is the Allan variance [35]. The Allan variance is defined as

$$\sigma_y^2(\tau) = \frac{1}{2}\overline{(\Delta y)^2} \tag{4.47}$$

where y is some random process, Δy denotes the first finite difference of the measurements of y. For example, if i denotes the i^{th} measurement of y, then $\Delta y = y_{i+1} - y_i$. In total, each adjacent finite difference of y is squared and then averaged over the data set and divided by 2. The divide by two causes this variance to be equal to the classical variance if the random components of y are taken from a random and uncorrelated set. This condition is only satisfied if the noise is white noise, i.e. thermal noise.

The advantage of this variance over the classical variance is that it converges for most of the commonly encountered kinds of noise, whereas the classical variance does not always converge to a finite value. Flicker noise and random walk noise are two examples which commonly occur in clocks and in nature where the classical variance does not converge.

Using the Allan variance measure, the PLL jitter over a time interval of ΔT can be given as [36]

$$\sigma_{\Delta T}^2 = \Delta T \sqrt{\frac{8}{f_0^2 \pi^2 (\Delta T)^2} \int_{f_{min}}^{f_{max}} S\phi(f_m) \sin^4(\pi f_m \Delta T) df_m} \qquad (4.48)$$

where f_0 is the VCO operating frequency, $S\phi(f_m)$ is the power spectral density of the noise shaped by one of the loop transfer functions of (4.42) – (4.46), f_m is the phase noise offset frequency.

The VCO in SoC processor PLLs usually has the largest contribution to the PLL's total jitter. For this reason, special attention is given to the VCO. As mentioned in the previous section, the noise introduced by the VCO, or is input referred to the VCO, is high-pass filtered by the PLL. This means that the long-term contribution of jitter by the VCO is negligible. Figure 4-22(a) shows the bode plot diagram of the free-running VCO phase noise and the VCO phase noise suppressed by the PLL's filtering response. As indicated, the VCO phase noise is suppressed up until the closed-loop 3-dB bandwidth, B_L, which is given by Equation (2.12). It continues to rise at 20dB/decade up until the loop bandwidth, then falls again at the same rate beyond the loop bandwidth.

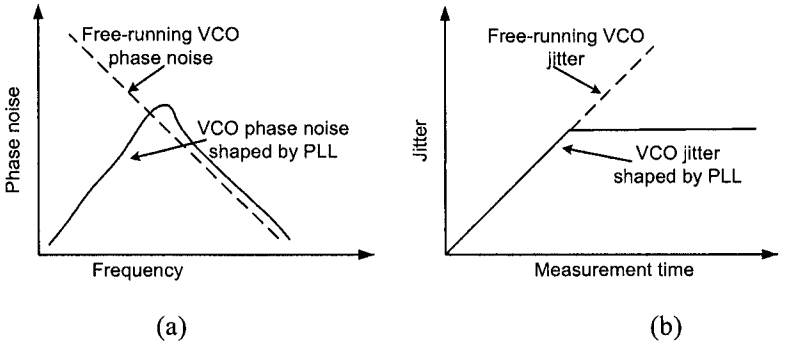


Figure 4-22. Shaping of VCO jitter by PLL in (a) frequency domain and (b) time domain

Substituting equation (4.46) into (4.48) gives the time domain jitter of PLL due to VCO noise. Figure 4-22(b) shows the Allan variance of the noise as a function of sampling time. As shown, the jitter increases until a time $1/B_L$. This point where the jitter reaches its maximum can be found by

integrating the VCO noise around the loop bandwidth, where it is most dominant. Assuming VCO jitter is dominated by upconverted thermal noise, the long-term jitter can be found to be

$$\sigma_{\Delta T}^2\Big|_{\Delta T\to\infty} = \left(\frac{N}{f_0^2}\right)\cdot\sqrt{\frac{1}{2\zeta\omega_n}} \tag{4.49}$$

where ω_n is the PLL's natural frequency given by (2.10) and ζ is the PLL's damping factor given by (2.11).

The flicker noise can also be of concern in PLLs. Since VCO noise is high-pass filtered by the PLL, it is expected that flicker noise (low-frequency noise) does not impact the total PLL jitter significantly. However, as transistor lengths of CMOS technologies shrink, the flicker noise corner frequency tends to be larger. For a typical 0.13um CMOS technology, it may be as large as several megahertz, well beyond the PLL's closed loop bandwidth. For this reason, flicker noise must be incorporated into jitter analysis.

For noise to be cyclostationary, its mean and autocorrelation function must be bounded and periodic. In the case of flicker noise, its mean and autocorrelation functions are *not* bounded and hence it cannot be considered to be cyclostationary noise. Instead, flicker noise is treated as correlated noise since it is quite possible to have the same error sample present for two or more cycles. Given that flicker noise is correlated noise, the variance of the jitter resulting from flicker noise can be given as

$$\sigma_{\Delta T}^2 = \kappa_{1/f}\cdot\Delta T \tag{4.50}$$

Figure 4-23 shows the effect of flicker noise on a free-running VCO in both the frequency domain and time-domain. If the flicker noise corner frequency is beyond the PLL's closed loop bandwidth, the effect of flicker noise on the PLL's total jitter is large. However, if the flicker noise corner frequency is below the loop bandwidth, then the flicker noise arising from the VCO is well suppressed.

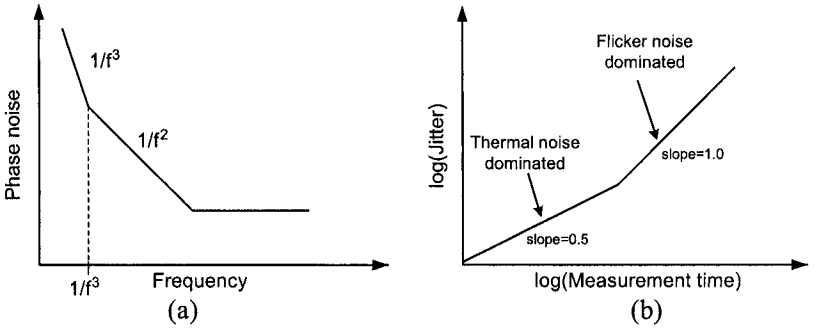


Figure 4-23. Effect of flicker noise on VCO (a) phase noise and (b) jitter

Other components of the PLL can also contribute to jitter [29]. Figure 4-24 shows the complete jitter plot over measurement period. For short measurement periods, the jitter from the VCO accumulates and hence increases the total PLL jitter. For moderate measurement periods, the jitter from the frequency dividers and charge pump dominate. For very long measurement periods, the jitter due to the external crystal oscillator starts to accumulate and increase jitter without bounds. For PLL characterization, however, excessive measurement periods can lead to incorrect results. As the VCO jitter increases, the middle portion of the curve shrinks as the first part of the curve expands to the right and increases in amplitude. An increase in jitter in the charge pump, frequency divider, or resistor noise in the loop filter, causes the amplitude of the middle portion of the curve to move upwards. Finally, an increase in external crystal oscillator noise can cause the last portion of the curve to shift to the left, shrinking the middle portion of the curve. Noise coupling to interconnect between the external crystal oscillator and the PLL can also cause the right portion of the graph to expand to the left.

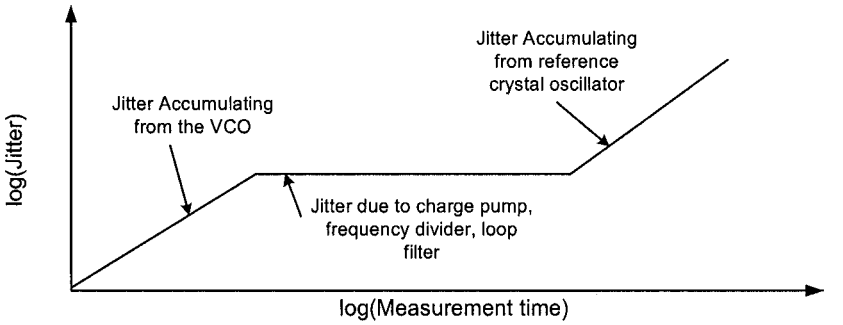


Figure 4-24. Complete jitter plot of all PLL noise sources

4.4.3 Effect of PFD Nonlinearities on Jitter

The phase-frequency detector (PFD) samples the feedback signal from the divider at a fixed comparison rate. The phase error between the two signals is then linearly translated into a voltage error to the VCO. There are two main sources of nonlinearities in the PFD. The first is the imbalance between UP and DOWN pulses. This would create a multiplicative factor associated with the phase error depending on the sign of the error. The other source of nonlinearity arises from the PFD's inability to distinguish between two very narrow pulses. This can be solved, to a large extent, by inserting delay in the feedback of the reset path in the PFD. Consider the two sets of waveforms shown in Figure 4-25. In Figure 4-25(a), no delay exists in the reset path. When the PLL is locked, or near locked condition, the PFD is

unable to generate a rail-to-rail signal for its UP output. This generates a net voltage on the VCO that is no longer linearly related to the phase error. This region is called the *dead zone* region. In Figure 4-25(b), the UP signal is guaranteed to be rail-to-rail by delaying the reset on both UP and DOWN by the same amount. This type of phase detector results in a more linear operation, and is often referred to as a type-4 PFD (see section 2.2.3).

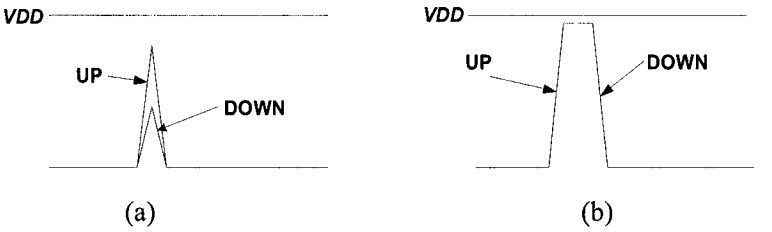


Figure 4-25. Transient response of PFD with different reset delays (a) no reset delay and (b) finite reset delay

One may wonder why is it so important to have a linear PFD? Consider the nonlinearity of the PFD expressed with the following Taylor series expansion

$$v(t) = a_1 \cdot \theta_e + a_1 \cdot \theta_e^2 + a_1 \cdot \theta_e^3 + \ldots \tag{4.51}$$

The input phase error is the difference between the reference signal (a fairly clean signal) and the feedback signal from the VCO divided down by the frequency divider. If the VCO contains any high frequency tones, these tones would appear in the θ_e term, which would then be filtered out by the low-pass filter response of the PLL. However, if the PFD contains any second order nonlinearities, this tone would be converted down to DC and would appear as a noise dependent static phase error at the PLL's output. If there are two high frequency tones spaced by Δf, then a third order nonlinearity would produce a tone at Δf. If Δf is less than the PLL's closed loop bandwidth, then this tone would not be filtered by the PLL and would appear as deterministic jitter at the output of the PLL.

The length of the reset delay is also important. On one hand it would be desirable to extend the length of the reset delay to preserve the linearity of the PFD. On the other hand, extending it both increases jitter as well as adversely affects the locking behavior of the PLL. During locked condition, the PFD outputs periodic UP and DOWN pulses of equal length. The jitter contribution of the charge-pumps, hence, is equal to the noise produced by both UP and DOWN current sources multiplied by the pulse duration and the noise transfer function from the charge pump to the PLL output, $H_{pfd}(s)$ (equation 4.43). The longer the reset pulse, the greater the jitter contribution

of the charge pump. Figure 4-26 shows the effect of reset delay on jitter in a PLL.

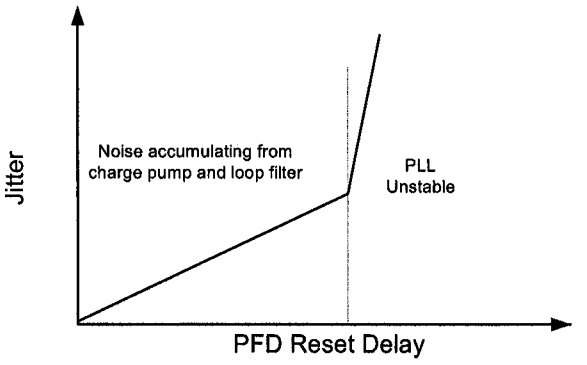


Figure 4-26. Effect of reset delay on jitter in a PLL.

Increasing the reset delay can also adversely affect the locking behavior of the PLL. The linear operating range of a PFD is effectively 4π radians [10]. This is assuming no reset delay. The reset delay directly subtracts from the linear range of the PLL, making it

$$T_{range} = 4 \cdot \pi - T_{reset} \tag{4.52}$$

This means as the PLL is locking it will exit the linear region more often (since it's shorter) and be more susceptible to cycle slipping. Figure 4-27 shows two PLL locking transient responses with different reset delays. Clearly, the longer the reset delay, the longer the lock time. In the limiting case, when the PFD reset delay exceeds ¼ the comparison period, the PLL can become unstable.

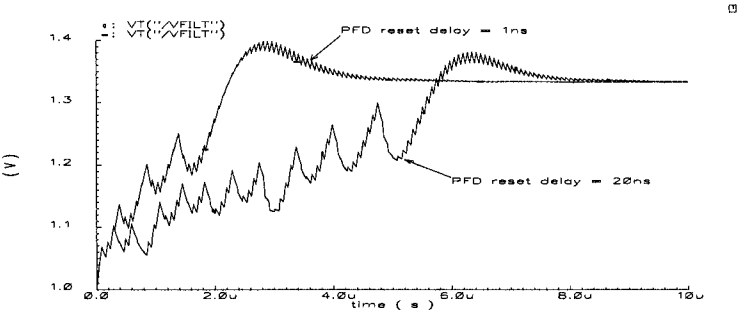


Figure 4-27. Effect of PFD reset delay on lock time

4.5 Coupling Noise Effects on Jitter

One major source of noise in microprocessor PLLs is externally coupled noise through the power supply and silicon substrate [20,21]. In this section, the mechanism of noise coupling into PLLs is examined in detail. The effect of factors such as technology, circuit topology, and loop parameter on noise coupling into PLLs is analyzed.

4.5.1 Power Supply Coupling Noise

In a SoC microprocessor, the PLL and the microprocessor may share the same power supply. For this reason, the power supply can be filtered in order to suppress as much noise as possible. This can be done through a voltage regulator as shown in Figure 4-28. The voltage regulator suppresses low frequency noise, whereas a passive RC filter suppresses higher frequency noise. Other possible topologies are discussed further in Chapter 5. In many cases, the power supply cannot be sufficiently cleaned to such a degree that the power supply noise can be ignored.

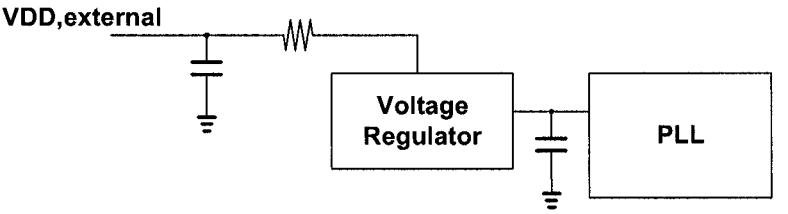


Figure 4-28. PLL power supply filtered through a voltage regulator.

Microprocessor transients can cause rippling on the supply voltage generating both systematic and accumulated phase error [17]. Power supply noise is a function of the number of digital switching circuit on the chip as well as the statistics of the switching behavior of each circuit. Two possible power supply rippling mechanisms are illustrated in Figure 4-29. Long on-chip power traces contain a finite amount of parasitic resistance. Even though this resistance may be a few ohms, it can still cause significant power supply drops. Consider the case when the microprocessor current suddenly jumps by 100mA. For an on-chip power route parasitic resistance of only 10Ω, the power supply will ripple by 1V, which cannot be tolerated for low-voltage applications. In addition to this ripple, the power supply feeding the PLL will be lower than the externally biased power supply by an amount equal to $I_{PLL} \cdot R_{par}$.

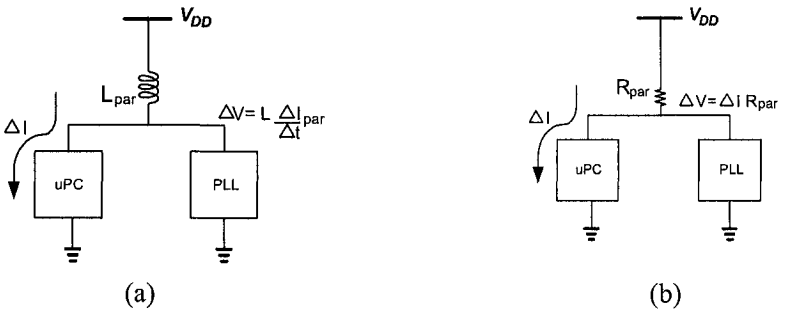


Figure 4-29. Power supply rippling by (a) L di/dt drops and (b) IR drops

The other mechanism shown in Figure 4-29(a) is significant during power transients. At steady-state, the power drop through the inductor is zero. Parasitic inductances result from board traces, packaging traces, and bond wire inductance can also come into play. At frequencies greater than 1GHz, parasitic inductances in on-chip power lines become significant [11].

One method of reducing the IR and $L\frac{di}{dt}$ drops is to add more supply pins. Figure 4-30 shows a typical package of a modern microprocessor with hundreds of power supply and ground pins. Extra pins, however, come at the expense of increased packaging and printed circuit board costs. Other methods include the use of low inductance package and the use of low noise logic family based on current steering instead of current switching.

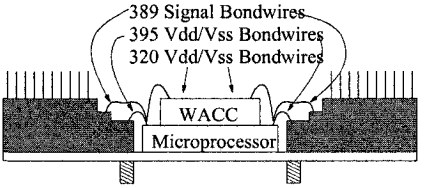


Figure 4-30. Typical package of a modern microprocessor

On-chip methods of reducing the effect of reducing the effects of power supply bounces may also be employed. Figure 4-31 shows a typical CMOS inverter with the parasitic inductances shown on both the power supply as well as ground lines. As the figure shows, the effect of parasitic inductance on the supply lines is glitching on the output waveform for both rising and falling edges of the input waveform.

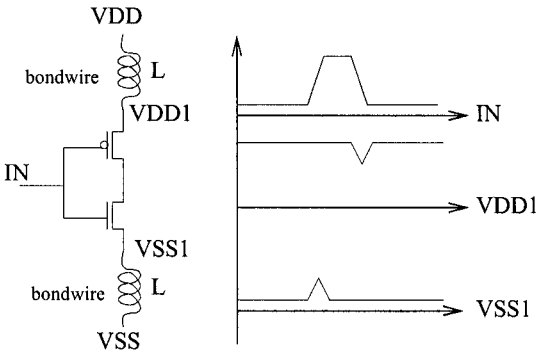


Figure 4-31. CMOS inverter with supply/ground bounces

One popular technique to reduce power supply bounces is to use decoupling capacitors inserted between the power supply and the ground line as shown in Figure 4-32(a). The capacitor acts in such a way that current spikes are absorbed and thus minimizing the magnitude of the ripple on the power supply line. Increasing the width of the power lines reducing the parasitic resistance associated with the line as well the increase the parasitic capacitance to ground. The former helps reduce IR drops and the latter helps reduce ground bounces cased by $L\frac{di}{dt}$ drops. In addition to the use of a decoupling capacitor, a linear regulator may be used as shown in Figure 4-32(b). The linear regulator acts as to limit the current going into the inverter and hence filtering any current spikes on the power supply.

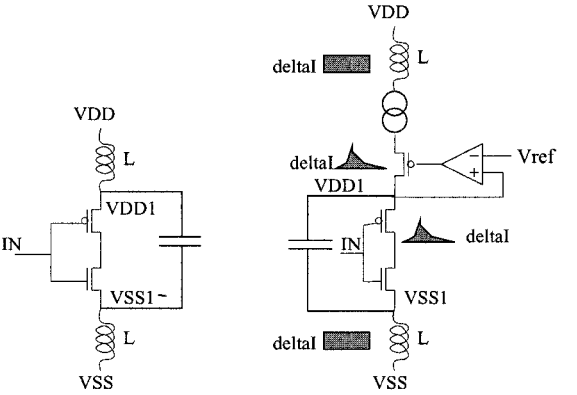


Figure 4-32. On-chip techniques for reducing power supply bounces: (a) decoupling capacitor (b) linear regulator

In order to analyze the effect of decoupling capacitors, consider the limiting case where the size of the inductance is infinite. The equivalent circuit is shown in Figure 4-33. A typical clocked system is assumed. C_d is the on-chip decoupling capacitor, C_s is the switched capacitance, Phi is the system clock, and C_n is the non-switched capacitance. Assuming this type of system. It can be shown that the voltage ripple seen as the switched capacitance is [12]

$$\Delta V = \frac{V_{DD} \cdot C_s}{2(C_s + C_d)} \tag{4.53}$$

This expression clearly shows that the supply voltage ripple can be directly reduced by the decoupling capacitor, C_d.

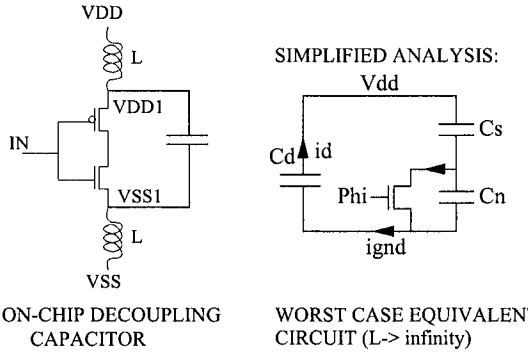


Figure 4-33. Equivalent circuit of circuit with decoupling capacitor

Now that the sources of power supply noise and methods to reduce it have been reviewed, one may ask how does it affect jitter in a PLL? In general, changing the supply voltage, alters the slew rate of the circuit. In the context of a ring oscillator, for example, power supply noise alters its zero crossings [26]. In the context of PFD, modifying the supply voltage changes its slew rate and delay time, and hence the steady-state on-time of the charge pumps varies. In the context of charge-pumps, modifying the supply voltage has an impact on its dynamic range. Supply noise coupling into the loop filter can modulate the VCO directly and hence generate jitter. In order to avoid this last effect, the loop filter and the current sources in the VCO are often referenced to the same supply voltage. In this way, the net ripple on the control voltage, as referenced to the VCO, is cancelled out.

In order to achieve good immunity to supply noise, the power supply rejection ratio (PSRR) of the sensitive analog circuits (charge pump and VCO) are maximized. For the charge pump, maximizing its PSRR amounts to maximizing the impedance from the supply to the output node. As shown

in Chapter 3, the DC PSRR can be maximized through the use of cascode devices. As the frequency of the supply noise increases, the impedance of the cascode drops (due to parasitic shunt capacitances) and the PSRR degrades [13]. The PSRR of the PLL's biasing reference circuit (PTAT or bandgap current reference) is also important since it will directly apply noise to the charge pump as well as the VCO. Techniques of enhancing the PSRR of the VCO are shown in Chapter 5.

One important measure in PLLs co-existing in digital noisy environments is its ability to prevent the transfer of supply noise into jitter. A commonly used figure of merit is "ps/mV", which is read as picoseconds of peak-to-peak jitter per mV of supply voltage changed. This is usually measured by stepping the supply voltage periodically and measuring the resulting peak-to-peak jitter. This is shown in Figure 4-34. The top plot is the supply voltage, and the bottom plot is the control voltage of the VCO. A value of 1.5ps/mV is typical of a modern PLL's performance under noise power supply conditions.

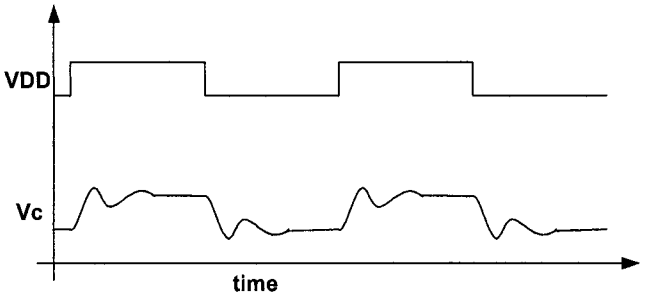


Figure 4-34. Evaluating performance of PLLs using "ps/mV" figure-of-merit

4.5.2 Substrate Coupling Noise

Another mechanism of noise coupling into the PLL is through the silicon substrate [22]. Although the noise coupling in this mechanism is less severe than the supply voltage, it is becoming more important as SoC processors become more integrated and are implemented in deeper submicron technologies.

Figure 4-35 shows the coupling mechanism from the substrate. The "n+" region to the left represents an NFET from nearby digital circuitry injecting noise into the substrate. The PLL's ground terminal is shown with the "p+" region to the right. The substrate resistance between the "n+" and "p+" regions is R1. A typical digital process has low substrate impedance in order to minimize the effect of latchup [14]. Assuming that the injected noise voltage magnitude is ΔV and the resistance from the PLL's ground terminal

on-chip to the ground terminal on the board is R_{PLLGND}, the injected voltage noise into the PLL is simply a resistor divider, and is given as

$$\Delta V_{PLLGND} = \frac{R_{PLLGND}}{R_{PLLGND} + R1} \Delta V \tag{4.54}$$

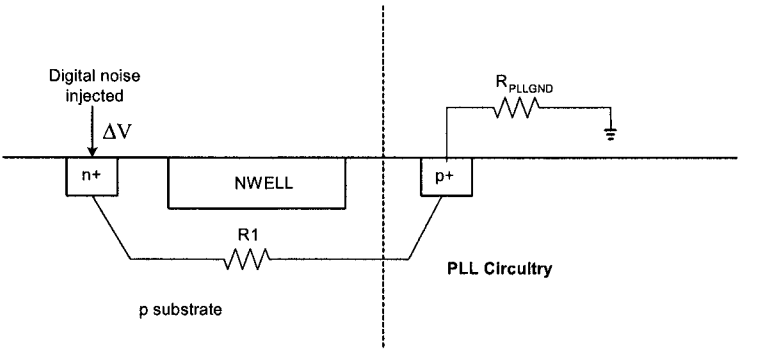


Figure 4-35. Substrate noise coupling mechanism

As the figure also shows, adding an NWELL barrier between the digital and analog circuitry accomplishes very little since most of the noise travels deep in the substrate, bypassing the NWELL barrier. One exception to this is if a special technology supporting deep WELL trench isolation, triple well, or silicon-on-insulator (SOI) technology is offered. Such technology options, however, are expensive and hence unattractive in the SoC processor market. The best method to provide isolation between analog and digital blocks is to increase the distance between the two blocks. Another method that is commonly used is to time-interleave the analog and digital blocks, such that when the analog circuitry is active the digital circuitry is off, and vice versa.

Methods of connecting the power and ground nets of a SoC PLL has a significant impact on its jitter performance. In [15], three possible power connections were examined as shown in Figure 4-36. Intuitively, one would expect that power supply configuration shown in Figure 4-36(c) gives the best result. However, it was found that the best combination was the power supply scheme shown in Figure 4-36(b). The reason for this lies in how the loop filter's capacitor is implemented. For best area density, the loop filter capacitors are implemented as NFET devices, where the gate capacitance is used as one terminal and the source, drain, and substrate terminals are all shorted together. The power configuration in Figure 4-36(c) would result in a situation where the source and drain are connected to the PLL's dedicated ground connection, whereas noise coupling into the substrate affects the

channel directly, resulting in a noise voltage appearing filter voltage. The power configuration shown in Figure 4-36(b), the substrate is explicitly shorted with the source and drain nodes. Hence, any noise injected across the channel is also injected to the source and drain of the transistor. Hence, the net differential voltage developing across the loop filter's capacitor is zero.

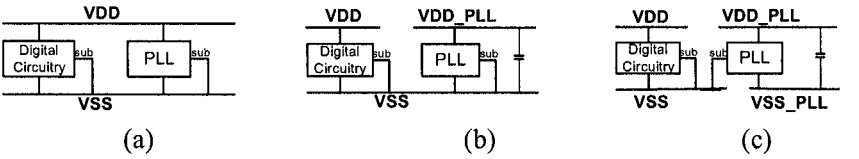


Figure 4-36. Three power supply schemes: (a) V_{DD}/V_{SS} shared, (b) separate V_{DD}, shared V_{SS}, (c) separate V_{DD}/V_{SS}

4.6 Summary

In this chapter, a summary of key concepts in jitter analysis in PLLs was given. First, the basic definitions of jitter were introduced and differentiated from one another. Next, the sources of jitter and the relationship between jitter and its frequency domain counterpart, phase noise, was detailed. This was followed by a discussion of jitter in VCOs. First, a derivation of Leeson's classical formula of phase noise and jitter was given. This was followed by a more recent time-varying analysis of VCOs. The discussion on jitter was then extended to include the entire closed-loop PLL.

One important source of jitter is supply and substrate coupled jitter. A detailed account of the mechanisms and sources of power supply noise was given. The effect of power supply noise on each critical PLL component was given. Techniques of reducing the effect of substrate noise on PLL jitter was also reviewed. This was followed by a recent analysis on the optimal choice of power supply connections for the PLL integrated in a digital process with digital circuitry.

REFERENCES

[1] T. Margeson, "Technology Symposiums," http://www.tektronix.com, 2003.

[2] M. Spiegel, *Mathematical Handbook of Formulas and Tables*, McGraw-Hill: New York, 1994.

[3] K. Wark and D. Richards, *Thermodynamics, 6th Ed.*, McGraw-Hill: New York, 1999.

[4] D. B. Leeson, "A Simple Model of Feedback Oscillator Noises Spectrum," *Proceedings of the IEEE*, vol. 54, pp. 329-154, Feb. 1966.

[5] L. S. Cutler and C. L. Searle, "Some aspects of the theory and measurement of frequency fluctuations in frequency standards," *Proceedings of the IEEE*, vol. 54, pp. 136-154, Feb. 1966.

[6] S. Haykin, *Digital Communications*, Wiley: New York, 1988.

[7] A. Hajimiri and T. Lee, *The Design of Low Noise Oscillators*, Boston: Kluwer Academic Publishers, 1999.

[8] J. Maneatis and M. Horowitz, "Precise delay generation using coupled oscillators," *IEEE J. of Solid-State Circuits*, vol. 28, no. 12, pp. 1273-1283, Dec. 1993.

[9] A. Hajimiri and T. Lee, "Design Issues in CMOS Differential LC Oscillators," *IEEE J. of Solid-State Circuits*, vol. 34, pp. 717-724, May 1999.

[10] D. Wolaver, *Phase-Locked Loop Circuit Design*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1991.

[11] E. Friedman, "Clock Distribution Networks in Synchronous Digital Integrated Circuits," Invited paper, *Proceedings of the IEEE,* no. 5, May 2001, pp. 3-14.

[12] P. Larsson and C. Svensson, "Noise in Digital Dynamic CMOS Circuits," *IEEE J. of Solid-State Circuits*, vol. 29, no. 6, pp. 655-662, June 1994.

[13] B. Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw-Hill: Boston, 2001.

[14] A. Bellaouar and M. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*, Boston: Kluwer Academic Publishers, 1995.

[15] P. Larsson, "Measurements and Analysis of PLL Jitter Caused by Digital Switching Noise," *IEEE J. of Solid-State Circuits*, vol. 36, no. 7, pp. 1113-1119, July 2001.

[16] J. McNeill, "Jitter in Ring Oscillators," *IEEE J. of Solid-State Circuits,* vol. 32, no. 6, pp. 870-879, June 1997.

[17] N. Barton, D. Ozis, T. Fiez, K. Marayam, "The Effect of Supply and Substrate Noise on Jitter in Ring Oscillators," *IEEE Custom Integrated Circuits Conference*, pp. 505-508, 2002.

[18] A. Hajimiri and T. Lee, "A General Theory of Phase Noise in Electrical Oscillators," *IEEE J. of Solid-State Circuits*, vol. 3, no. 2, pp. 179-194, February 1998.

[19] A. Hajimiri, "Noise in Phase-Locked Loops," *IEEE Custom Integrated Circuits Conference*, pp. 1-6, 2001,

[20] P. Heydari and M. Pedram, "Jitter-Induced Power/Ground Noise in CMOS PLLs: A Design Perspective," pp. 209-213, 2001.

[21] F. Herzel and B. Razavi, "A Study of Oscillator Jitter Due to Supply and Substrate Noise," *IEEE Trans. On Circuits and Systems II*, vol. 46, no. 1, pp. 56-62, January 1999.

[22] R. Welch and A. Yang, "Substrate Coupling Analysis and Simulation for an Industrial Phase-Locked Loop," *IEEE Int'l Symposium on Circuits and Systems*, vol. VI, pp. 94-97, 1998.

[23] L. Dai and R. Harjani, "Design of Low-Phase-Noise CMOS Ring Oscillators," *IEEE Trans. On Circuits and Systems II*, vol. 49, no. 5, pp. May 2002.

[24] A. Hajimiri, S. Limotyrakis, T. Lee, "Jitter and Phase Noise in Ring Oscillators," *IEEE J. of Solid-State Circuits*, vol. 34, no. 6, pp. 790-801, June 1999.

[25] A. Hajimiri, S. Limotyrakis, T. Lee, "Phase Noise in Multi-Gigahertz CMOS Ring Oscillators," *IEEE Custom Integrated Circuits Conference*, pp. 49-52, 1998.

[26] B. Kim, T. Weigandt, P. Gray, "PLL/DLL System Noise Analysis for Low Jitter Clock Synthesizer Design," *IEEE Int'l Symposium on Circuits and Systems*, pp. 31-34, 1994.

[27] B. Razavi, "A Study of Phase Noise in CMOS Oscillators," *IEEE J. of Solid-State Circuits*, vol. 31, no. 3, pp.331-343, March 1996.

[28] C. W. Zhang and L. Forbes, "Timing Jitter in a 1.35-GHz Single-Ended Ring Oscillator," *IEEE Int'l Symposium on Circuits and Systems*, vol. 3, pp. 308-311.

[29] M. Mansuri and C. Yang, "Jitter Optimization Based on Phase-Locked Loop Design Parameters," *IEEE J. of Solid-State Circuits*, vol. 37, no. 11, pp. 1375-1382, November 2002.

[30] Y. Chen, S. Koneru, E. Lee, R. Geiger, "Effects of Random Jitter on High-Speed CMOS Oscillators," *IEEE Int'l Symposium on Circuits and Systems*, vol. 3, pp. 176-179.

[31] W. Liu, *MOSFET Models for SPICE Simulation*, New York: John Wiley & Sons, Inc. 2001.

[32] J. Ingino, "A 4-GHz Clock System for a High-Performance System-on-a-Chip Design," *IEEE J. of Solid-State Circuits*, vol. 36, no. 11, pp. 1693-1698, November 2001.

[33] D. Ham and A. Hajimiri, "Concepts and Methods in Optimization of Integrated LC VCOs," *IEEE J. of Solid-State Circuits*, vol. 36, no. 6, pp. 896-909, June 2001.

[34] T. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge: University Press, 1998.

[35] D.W. Allan, "Time and Frequency (Time-Domain) Characterization, Estimation, and Prediction of Precision Clocks and Oscillators," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, UFFC-34*, pp. 647-654, 1987.

[36] D. Sahu, "A Completely Integrated Low Jitter CMOS PLL for Analog Front Ends in System on Chip Environment," *VLSI Design Conference*, 2002.

Chapter 5

LOW-JITTER PLL ARCHITECTURES

5.1 Introduction

In the previous chapters, a generic charge-pump PLL architecture was assumed. Circuit techniques as well as jitter analysis were conducted based on this architecture. In this chapter, a variety of state-of-the-art PLL architectures are reviewed. Each have their merits and disadvantages in terms of jitter, power consumption, and area. Each class of architectures is analyzed individually, including applicable circuit implementations.

5.2 Differential PLL Architecture

The first architecture to be analyzed is the differential PLL architecture. Two variants of this architecture are analyzed below. A conventional topology is first described. Relevant circuit techniques required to implement a differential PLL architecture are also discussed. A second, higher performance differential PLL architecture is then described.

5.2.1 Conventional Topology

A differential PLL topology is shown in Figure 5-1 [1]. As the figure shows, every component in the PLL is implemented differentially, even the loop filter and VCO. If implemented correctly, the differential PLL architecture should have excellent common mode rejection of noise and co-exist quite well with nearby noisy digital circuitry. Implementing the loop filter differentially is also advantageous in term of area, since the loop filter

can be implemented differentially and thus reducing capacitor size by one-quarter.

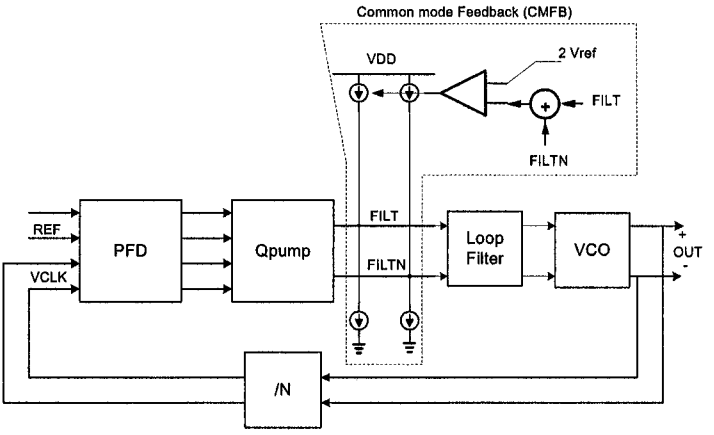


Figure 5-1. Differential PLL architecture

As shown in Figure 5-1, a common mode feedback (CMFB) circuit is required. The function of the CMFB is to ensure that the common mode voltage, or average voltage, of the two loop filter lines is fixed to a certain voltage reference. The first part of the CMFB circuit is a circuit that senses the common mode voltage. The second part of the CMFB circuit drives the loop filter voltages to either a higher or lower potential such that their average is fixed to a certain value, Vref. This is done by controlling the bottom current sources as shown in Figure 5-2 such that the net current is either added or subtracted from the loop filter. The CMFB circuit operates continuously. Alternatively, the top current sources can also be controlled by the CMFB circuit. The two resistors used in Figure 5-2 are sized large enough such that there is minimum differential leakage current.

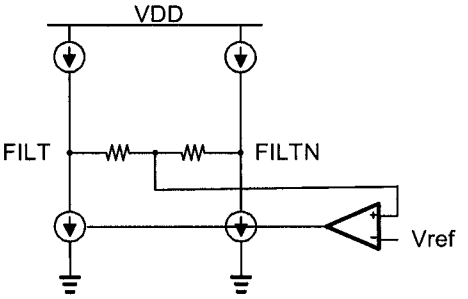


Figure 5-2. Sample CMFB circuit

Mismatch between the bottom and top current sources is compensated for by the feedback in the CMFB circuit. The larger the gain in the feedback, the smaller the error between the top and bottom currents at steady state. Since this mismatch is common mode, it does not disturb the loop dynamics. However, M\mismatch between the two bottom current sources or the top two current sources does have a large impact on loop dynamics. Mismatch in this case has the effect of creating a differential current error, which causes a differential voltage to develop across the loop filter. Since this current mismatch, ΔI, is continuous and it can only be corrected once every reference period, it translates into a reference spur. This increases the deterministic jitter significantly. Even for small current mismatches, the reference spur can be quite large. Figure 5-3 shows the dependency of the reference spur for a given current mismatch ΔI. If ΔI becomes comparable to the charge pump current, I_p, then this may cause the PLL to lose lock or to false lock to a different frequency. A reference spur greater than -40dBc causes noticeable increase in jitter. A reference spur greater than -20dBc may cause the PLL to false lock.

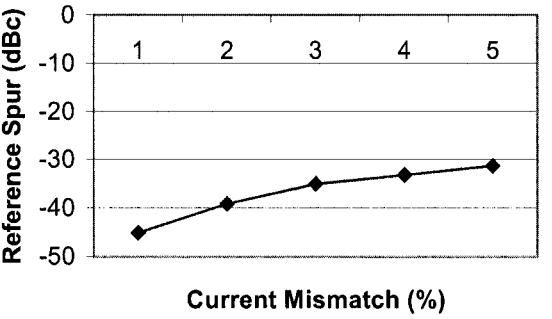


Figure 5-3. Reference spur versus mismatch current ΔI

Current mismatch results from device mismatch in the current mirrors used to implement the current sources of the CMFB shown in Figures 5-1 and 5-2. One method to analyze the effect of device mismatch is to perform a sensitivity analysis of a transistor operating in the saturation region. For a long channel device in saturation, the current is given as

$$I_{DS} = \frac{1}{2}\mu \cdot C_{ox} \cdot \frac{W}{L}(V_{GS} - V_T)^2 \tag{5.1}$$

First, let us analyze the sensitivity of the current to threshold voltage mismatches. Differentiating equation (5.1) with respect to V_T yields

$$\frac{\Delta I}{\Delta V_T} = \mu \cdot C_{ox} \cdot \frac{W}{L}(V_{GS} - V_T) \tag{5.2}$$

Dividing (5.2) by (5.1) gives the variation of current as a fraction of the total current and is expressed as

$$\frac{\Delta I}{I_{DS}} = \frac{2 \cdot \Delta V_T}{V_{GS} - V_T} \tag{5.3}$$

where ΔV_T is given as [2]

$$\Delta V_T = \frac{A_{VT}}{\sqrt{W \cdot L}} \tag{5.4}$$

where A_{VT} is the statistical variation of the threshold voltage in units of mV·μm. A typical value of A_{VT} is 10mV·μm. Equation (5.4) indicates that the larger the device the less the mismatch. Another important fact is the current mismatch due to V_T mismatches, depends on the overdrive voltage, V_{GS}-V_T, of the bias transistor. For typical values of ΔV_T = 10mV and an overdrive voltage of 200mV, the current mismatch can be as high as 10%, yielding a large reference spur as indicated in Figure 5-3. For this reason, V_T mismatches usually result in the largest component of current mismatches in current sources.

Differentiating (5.1) with respect to transistor width W gives

$$\frac{\Delta I}{\Delta W} = \mu \cdot C_{ox} \cdot \frac{1}{L}(V_{GS} - V_T)^2 \tag{5.5}$$

Dividing this by (5.1) shows that the variation of current for a transistor width mismatch is simply given as

$$\frac{\Delta I}{I} = \frac{\Delta W}{W} \tag{5.6}$$

where ΔW is a statistical measure of the variation of the transistor width. Performing the same sensitivity analysis on transistor length shows that the current mismatch due to transistor length variation is

$$\frac{\Delta I}{I} = -\frac{\Delta L}{L} \tag{5.7}$$

where ΔL is the statistical variation of transistor length, L. The transistor length variation can be reduced by using multiple transistors in parallel giving the same effective W/L ratio. For example, if a certain transistor W/L

ratio is desired, the transistor can be implemented as n transistors for width W and length n·L connected in parallel. The effective variation in transistor length can be reduced by [2]

$$\Delta L = \frac{\Delta L_o}{\sqrt{n}} \tag{5.8}$$

where ΔL_o is the statistical variation of length of a transistor of width W_o.

The current mismatch ΔI caused by any of the above mismatch mechanisms is injected directly into the loop filter. This mismatch causes a reference spur with magnitude [3]

$$\text{Spur} = \frac{\frac{1}{2}\Delta V \cdot K_V}{2 \cdot F_{REF}} \tag{5.9}$$

where F_{REF} is the reference frequency, K_V has units of Hz/V, and ΔV is the change in loop filter voltage due to differential current ΔI. ΔV is given as

$$\Delta V = \frac{\Delta I}{F_{REF} \cdot C_T} \tag{5.10}$$

where C_T is the total single-ended loop filter capacitance. Substituting (5.10) back into (5.9) reveals that the spur level can be expressed as

$$\text{Spur} = \frac{\frac{1}{2}\Delta I \cdot K_V}{2 \cdot F_{REF}^2 \cdot C_T} \tag{5.11}$$

This shows that the magnitude of the reference spur has a strong dependency on the phase detector comparison frequency.

The disadvantage of the CMFB circuit shown in Figure 5-2 is that very large resistors (in the order of megaohms) may be needed in order to differential current leakage. This may be resolved by using source followers between the loop filter nodes and the common mode feedback circuit as shown in Figure 5-4. In this case, the resistors can be greatly reduced without any disturbance to the filter nodes.

There are, however, certain disadvantages of this topology. This topology consumes more power since extra source follower branches are needed. Secondly, the minimum loop filter is now constrained by the CMFB circuit, which now consists of the headroom required for the current source in addition to the threshold voltage of the source follower device. It is important to note that the source follower device suffers from the body effect, which increases the effective V_T by an amount proportional to the source voltage.

Yet another CMFB topology that avoids resistors altogether is shown in Figure 5-5. This topology uses two differential pairs that steer a current I_B either through a diode-connected transistor producing the control voltage

V_{CTRL} or through a dummy branch. Consider the scenario where the common mode voltage of the loop filter is higher than V_{REF} and both FILT and FILTN are higher than V_{REF}. In this case, the current dumped into MN1 is much larger than that of MN2 and the control voltage is increased. Through the error amplifier, the top currents are reduced. This would draw current away from both FILT and FILTN and the common mode voltage is reduced.

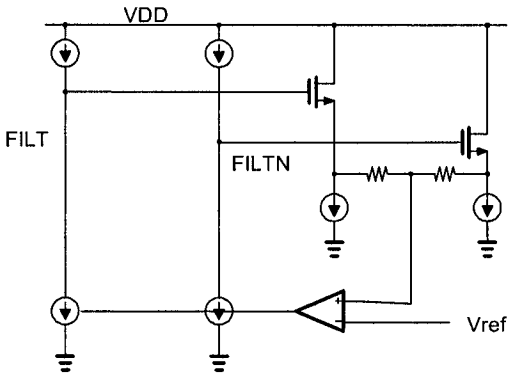


Figure 5-4. CMFB circuit with source followers

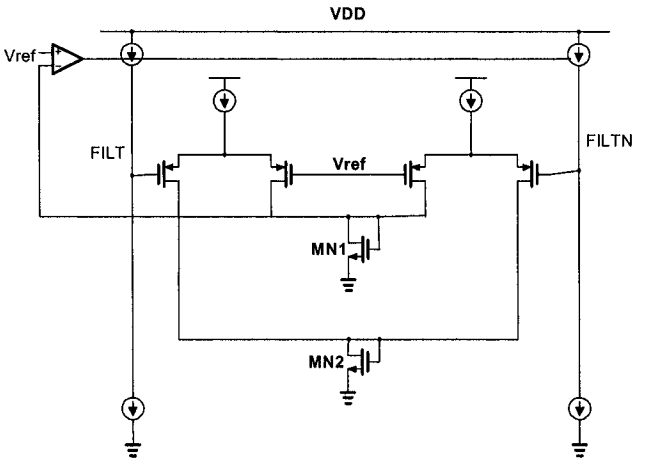


Figure 5-5. Resistorless CMFB circuit

Another important effect in CMFB circuits when used in differential PLLs is the output impedance of the current sources. Consider the scenario when FILT and FILTN are far apart and simple current mirrors of the type shown in Figure 3-6 are used. The differences in V_{DS} of the top or bottom

pair of transistors would create a difference in the currents supplied. In other words, the reference spur would depend on the output frequency. In order to prevent this, high output impedance current sources are desired. Again, a very small mismatch between the pair of current sources due to V_{DS} differences can cause large output spurs and hence an increase in deterministic spurs. Cascode current mirrors can reduce the size of this reference spur.

In the case of perfect matching between all current sources, noise performance of the PLL is still affected negatively by the CMFB. As mentioned earlier, the CMFB is always enabled. Since the four current sources of the CMFB are attached directly to the loop filter voltage, they contribute to phase noise on the PLL output with the same transfer function of equation (4.43). Using this equation and a first order loop filter, the output transfer function due to current noises I_n can be given as

$$\overline{\theta_e^2(s)} = \left\| \frac{2\pi \cdot K_V \cdot R \cdot \left(s + \frac{1}{RC}\right)}{s^2 + \frac{K_V \cdot I_p \cdot R}{N} s + \frac{K_V \cdot I_p}{N \cdot C}} \right\|^2 \cdot \overline{I_n^2} \tag{5.12}$$

where K_V has units of Hz/V. This is a bandpass filter with a single zero and two poles. Although this is the same transfer function as from the charge pump to the VCO output, its behavior is somewhat different. In evaluating the effect of charge pump noise and the VCO output, changing the charge pump current affects the loop dynamics and noise performance concurrently. The loop bandwidth increases with increasing charge pump current. Also, as the current is increased, the current noise of the charge pump circuit also increases. Assuming that the same current density is maintained, an increase in charge pump current can only occur with an increase in W/L. Since the transistor transconductance g_m is linearly proportional to W/L, it follows from equation 3.18 that the current density i_{ds}^2 also increases linearly with current. This is assuming that the noise is dominated by thermal noise. Also, comparing (5.12) with the standard equation of a PLL from PFD input to VCO output (equation 2.9), it is clear that the DC value of the transfer function of (5.12) is less than (2.9) by I_p, the charge pump current. This means that the output spectral density of any close-in phase noise is suppressed by $\frac{1}{I_p^2}$ times the input power spectral density, which is proportional to I_p^2. This means that the overall close-in phase noise is improved by I_p.

The increase in CMFB current, however, does not result in any improvement in close-in phase noise. On the contrary, larger current only

worsens the PLL's jitter performance. Increasing the charge pump current does not help in reducing the magnitude of this spur. Only increasing the size of the capacitors or reducing the VCO gain can reduce the size of this spur. One way to reduce the noise contribution of the CMFB circuit is to reduce the size of its the current sources. This, however, comes at the expense of longer PLL startup time.

5.2.2 Low-Noise Differential PLL Topology

Another solution to reduce the noise contribution of the CMFB circuit, is to turn off its current source dynamically. A common mode circuit that performs this function is shown in Figure 5-6 [4]. As in the common mode circuit shown in Figure 5-5, the circuit first operates by generating a voltage across a diode connected transistor that is proportional to the sum of the differential filter node voltages (left branch of Figure 5-6). This voltage is then compared with a reference voltage through another PFET input differential pair with diode connected NFET loads (middle branch of Figure 5-6). As long as the common mode level is less than the reference voltage, an equal amount of current is drawn from both FILT and FILTN. It is assumed here that the charge pump always pumps up (either to FILT or FILTN), so there is no need for a complementary pull-up CMFB circuit.

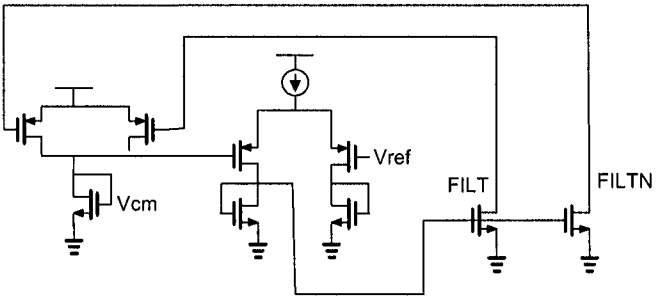


Figure 5-6. Common-mode feedback circuit with dynamic tail current

At steady state, no current is drawn from the loop filter. This greatly helps in reducing spurs generated by mismatches in currents in any one of the common mode feedback circuits previously discussed. It should be noted, however, that mismatch between any of the diode connected NFET devices in the differential pair can result in a shift in the common mode voltage, which can be tolerated. However, any mismatch between the two NFETs that pull current away from FILT and FILTN (right side of Figure 5-6), can result in differing amount of current being pulled from the filter nodes. This means that a reference spur is still possible with this architecture. However, since the common mode feedback current is very

low at steady-state, the resulting reference spur is much smaller than in a conventional common-mode feedback circuit.

5.3 Supply Voltage Regulated PLL Architectures

The main motivation behind a differential PLL architecture is to reduce the effect of power supply bounces on the PLL jitter performance. As shown above, the differential architecture inherently has higher reference spurious performance than a single-ended PLL topology. An alternative to a differential PLL architecture is to use voltage regulators to isolate the power supply fluctuations from the PLL. Typically, the VCO and the on-chip current references are isolated from the power supply noise.

5.3.1 Basic Concept

The basic idea behind supply voltage regulation is to enhance the output impedance looking into the supply voltage relative to ground. In mathematical terms, the power supply rejection (PSR) can be expressed as

$$PSR = \frac{z_{gnd} + z_{supply}}{z_{gnd}} \tag{5.13}$$

where z_{gnd} is the impedance looking into the ground terminal and z_{supply} is the impedance looking into the power supply terminal. Ideally, the regulated line voltage would not change as the supply voltage is varied. Figure 5-7 shows an example of how a regulated line would vary as the supply voltage is varied. The PSR is proportional to the inverse of the slope of the curve. The smaller the slope, the better the PSR.

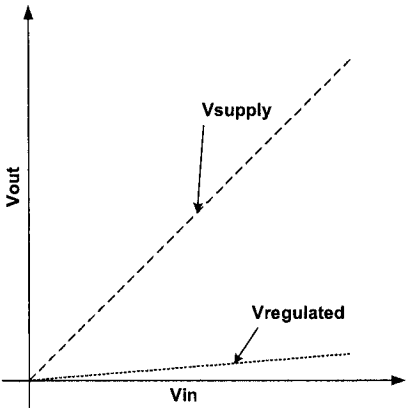


Figure 5-7. Variation of regulated line with power supply

Another often used metric is the power supply rejection ratio (PSRR). The PSRR is defined as the ratio of the transfer function from input to output to the transfer function from power supply to the output [32]. In case of voltage regulators, the input-output transfer function is unity; therefore, the PSRR is simply the inverse of the transfer function from the input power supply to the output regulated supply. In other words, the PSRR and the PSR are equivalent in the special case of voltage regulators.

5.3.2 Regulator Topologies

There are a variety of methods in which the power supply can be regulated. The simplest method is to use a cascode PFET buffer. This can be easily used in an ICO as shown in Figure 5-8. The V2I shown is also cascaded with a PFET current mirror. The PFET current mirror then supplies the current into the ICO, which in this case is a single-ended three stage CMOS inverter ring.

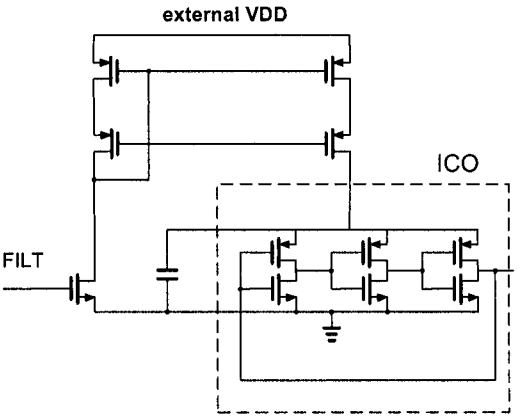


Figure 5-8. ICO with cascaded PFET current mirror

As shown in chapter 4, this ICO topology as a sharp and symmetric ISF, but suffers from poor power supply rejection. Including a PFET cascode device improves its DC power supply rejection by an amount proportional to the cascode's output resistance which is

$$R_{out} = g_{m2} \cdot R_{DS1} \cdot R_{DS2} \tag{5.14}$$

Typically, this translates to 10-20dB of improvement in power supply rejection ratio (PSRR).

The output impedance can be enhanced using an amplifier in the feedback as shown in Figure 5-9. Assuming that amplifier A is an inverting amplifier, it can be shown that the dc output resistance of the enhanced cascode structure is

$$R_{out} = A \cdot g_{m2} \cdot R_{DS1} \cdot R_{DS2} \tag{5.15}$$

In contrast with equation (5.14), the output impedance is enhance by a factor of A, where A is the dc gain of the feedback amplifier.

Amplifier A can be implemented by a single transistor as shown in Figure 5-10. The current source I is chosen in such a way that the feedback PFET transistor is always in saturation. The transconductance, g_m, of the PFET device is also affected by the current source I. The dc output resistance of the single transistor enhanced output impedance regulator can be given as

$$R_{out} = (g_{m3} R_{DS3}) g_{m2} R_{DS2} R_{DS1} \tag{5.16}$$

This type of regulation has been used successfully in [5] to regulate the VCO power supply.

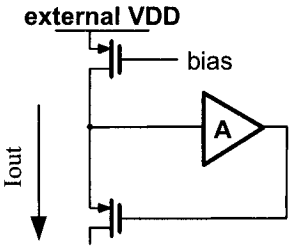


Figure 5-9. Enhanced output impedance with cascode

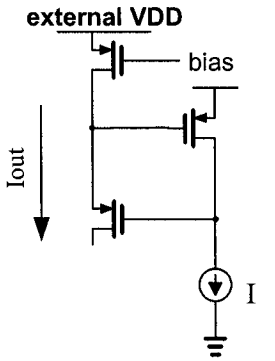


Figure 5-10. Single-FET enhanced output impedance regulation

5.3.2.1 Linear Voltage Regulator

A common form of regulation is linear voltage regulation, also known as low dropout (LDO) regulator. A generic linear regulator is shown in Figure 5-11. A feedback error amplifier is used to compare the output voltage with a reference voltage. The output voltage is given as

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) \cdot V_{ref} \tag{5.17}$$

where V_{ref} is usually generated by a bandgap reference. Any variation in the external supply reference is corrected by the feedback action of the control loop. Since the control loop requires a transient time for correction, the correction capability degrades with frequency. This means that the PSRR degrades with frequency.

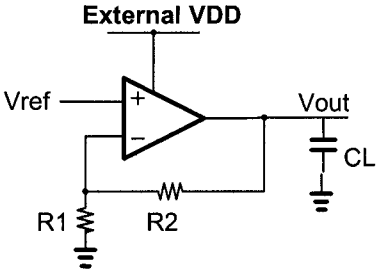


Figure 5-11. Linear Voltage Regulation

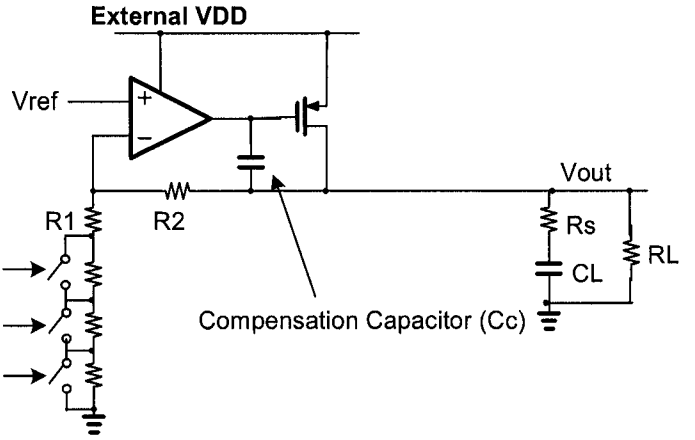


Figure 5-12. Practical linear voltage regulator

In a practical voltage regulator R2 and R1 are selected to give the desired output voltage and are sometimes made to be software programmable through FET switches. Figure 5-12 shows a voltage regulator with

programming switches on resistor R1. The error amplifier is also implemented as a two stage amplifier, with the second stage being a PFET usually operated in the triode region. A triode region is usually preferred since it offers a lower voltage drop across the PFET diode (hence the name low-dropout – LDO – regulator). Since the amplifier is a multistage amplifier (at least two stages, the second one being the PFET device), stability may be an issue (it is assumed that each amplifier contributes to at least one pole). Compensation capacitor C_C, shown in Figure 5-12, may be required to stabilize the regulator. Stability is usually worst for small loads (largest R_L). For this reason, it is common to have a bleeding current in parallel with the load to ensure a minimum current draw required for stability.

It is important to note that when the voltage regulator is supplying large current, the output load is no longer simply capacitive. The current draw is modeled as a shunt resistor in parallel with capacitor C_L. Since C_L and R_L are effectively on the feedback node of the control loop, the amount of current drawn by the voltage regulator affects its stability.

Another important point is that resistor R1 affects the efficiency of the regulator. If R1 is small, a significant amount of current from the PFET load device is wasted in the regulator itself, instead of being drawn into the load. For this reason, R1 is usually sized as large as possible.

For large C_L and moderate current draw, the dominant pole can be given as

$$f_1 = \frac{1}{2\pi C_L R_{out}} \tag{5.18}$$

where R_{out} is the output impedance of the PFET device in parallel with R_L. This means that for stable voltage regulator operation, the PSRR will degrade for frequencies beyond f_1.

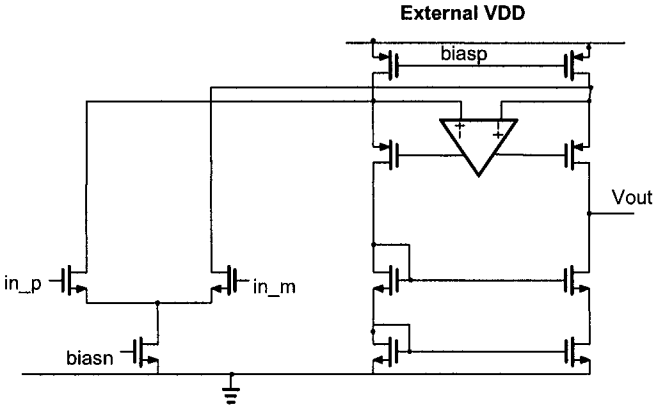


Figure 5-13. Folded-cascode gain-boosted op amp

Closer examination of the output load in Figure 5-12 reveals that it is similar to the passive loop filter used in charge-pump PLLs. R_S is the equivalent series resistance on the supply line. Although R_S is small, it has a very significant impact on loop dynamics since it introduces a zero. Since the parasitic resistance of the supply line is difficult to determine at design time, a tolerance range is usually estimated for voltage regulator design purposes.

One way to increase the gain of the error amplifier is to use a gain boosting stage. One such configuration is shown in Figure 5-13 [6,7]. The gain boosting stage ensures that the cascode devices remain deep in saturation under process, temperature, and supply voltage fluctuations. This type of amplifier has been used as part of a voltage regulator for a PLL and the reported dc gain was in excess of 70dB. The gain boosting stage was responsible for at least 10dB of this gain.

Figure 5-14 shows how a voltage regulator is used in a PLL's ICO. A typical voltage regulator design usually improves dc PSRR by 20dB to 50dB. The rate depends on the gain achievable in the error amplifier. The closed loop bandwidth of the regulator is rarely above 10MHz. This means that high frequency noise injected into the ICO from the power supply is not rejected even if a voltage regulator is used.

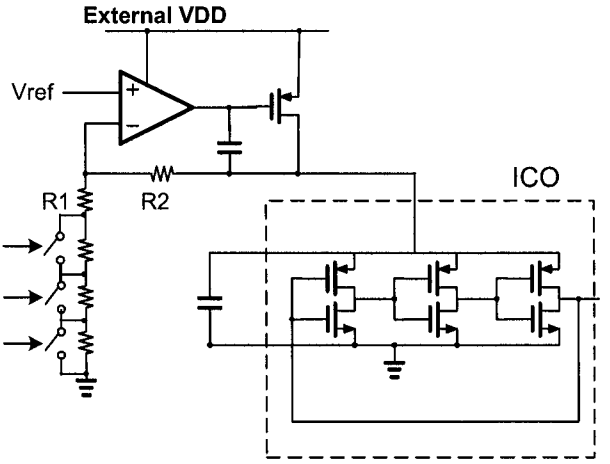


Figure 5-14. Voltage Regulator used in an ICO

5.3.2.2 Source Follower Voltage Regulator

Linear regulators are widely used in low-jitter microprocessor PLLs. Their main disadvantage is large area and low frequency response. Another type of voltage regulation that has been used successfully in microprocessor PLLs is a source follower voltage regulator.

An example of how a source follower circuit can be used in a ICO is shown in Figure 5-15. An NFET device is inserted between the external power supply and the ICO. The NFET is always in saturation, meaning that its output resistance is high. This high output resistance is what enables the source follower to act as a voltage regulator. Furthermore, the effective ICO supply voltage, to the first order, does not depend on the external supply voltage. Instead, it is always a $V_{GS,N}$ volts less than the bias voltage. If the bias voltage is a power supply independent voltage, then so is the ICO supply voltage. Due to the early effect, there is some dependence of the ICO effect supply voltage of $V_{DS,N}$ and hence the external supply voltage.

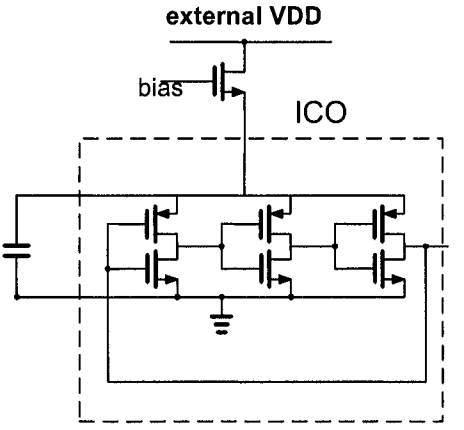


Figure 5-15. Use of a source follower for voltage regulation in ICO

One advantage of a source follower voltage regulator over a linear regulator is its wide bandwidth and inherent stability. As in a linear regulator, the load on the regulator can be modeled as a capacitor with series resistance and another parallel resistor, as shown in Figure 5-16.

As equation (5.13) shows, a circuit's power supply rejection depends on the ratio between the impedance seen to the ground and the impedance seen to the supply. The impedance seen to the ground is simply the equivalent load of the source follower voltage regulator. Assuming the load shown in Figure 5-12, the impedance to ground is given as

$$Z_{gnd}(s) = \frac{R_L R_S \cdot \left(s + \frac{1}{R_S \cdot C_L}\right)}{(R_S + R_L) \cdot \left(s + \frac{1}{C_L \cdot (R_S + R_L)}\right)} \tag{5.19}$$

At DC, it is simply R_L. At very high frequency, the impedance to ground is the equivalent parallel resistance of R_L and R_S.

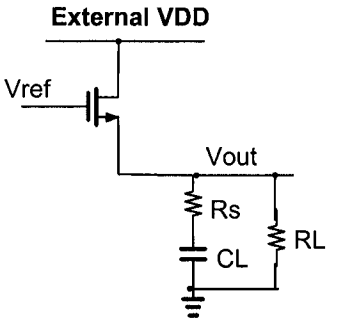


Figure 5-16. Source follower regulator with equivalent load circuit

When the input noise is injected through the supply, the source follower NFET is configured as a common gate amplifier [49]. The small signal impedance seen from the regulated line to the external supply line is given as

$$Z_{supply}(s) = \frac{1}{C_{gs} \cdot \left(s + \frac{g_m + 1/r_o}{C_{gs}} \right)} \tag{5.19}$$

where C_{gs}, g_m, and r_o are the gate-to-source capacitance, transconductance, and output resistance of the NFET source follower transistor, respectively. This equation neglects the body effect of the source follower transistor. Assuming an AC input from the supply and the output being the regulated supply voltage, an equivalent model incorporating both the NFET device and the equivalent load is shown in Figure 5-17. As the figure shows, there are three capacitors and one resistor in series with a capacitor. This means that there are three poles and one zero, respectively. One of the poles, introduced by C_{gd}, can be neglected for all practical purposes since it is only significant at frequencies beyond 1GHz.

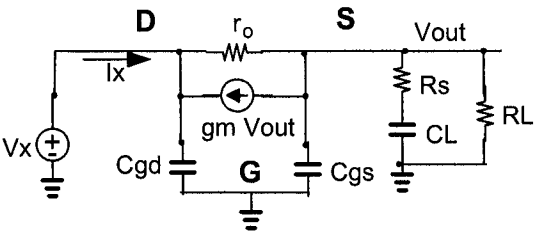


Figure 5-17. Equivalent high-frequency model of NFET voltage regulator

The PSR is simply the inverse of the transfer function of the model shown in Figure 5-17. Using equation (5.13), and the equivalent model of Figure 5-

17, it follows that the source follower's power supply rejection is given as

$$PSR_{NFET,SF}(s) = \frac{Z_{gnd} - g_m r_o Z_{gnd} - (sC_{gs}Z_L + 1)r_o}{Z_{gnd}} \tag{5.20}$$

where Z_{gnd} is the equivalent impedance of the load shown in Figure 5-16 and is given by equation (5.17). This is a two zero and a single pole system. Figure 5-18 shows a simulation plot of the frequency response of the source follower with the load shown in Figure 5-16. An AC input was applied to the supply and the output is plotted. This is effectively the inverse of the power supply rejection (PSR). The plot demonstrates a two pole single zero system, which corresponds to the inverse of the PSR frequency response equation (5.20). It is interesting to note that the PSR is constant for a large frequency range, then improves at high frequencies. This improvement seems to be contradictory to what can be measured, and will be explained shortly.

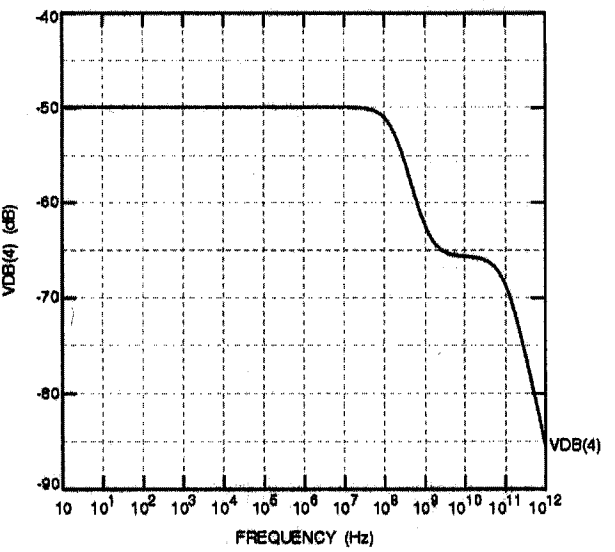


Figure 5-18. Frequency response of source follower with typical load

The DC value of the PSR is given as

$$PSR_{NFET,SF}(dc) = \frac{R_L - (g_m R_L + 1)r_o}{R_L} \tag{5.21}$$

This means that the DC power supply rejection (which is the worst case power supply rejection) depends on the g_m and output resistance of the

NFET device as well as the equivalent load resistance, which is inversely proportional to the average current draw.

Figure 5-19 shows the simulated and predicted DC PSR as a function of both R_L and g_m. The transcoductance is controlled through the width of the NFET device. As the figure indicates, the DC PSR is fairly constant for a wide range of R_L and g_m. This is because, as equation (5.20) indicates, that for relatively large values of R_L is PSR remains fairly constant. As R_L shrinks in size, the numerator DC PSR becomes less dependent on R_L and hence the denominator causes the PSR to increase as R_L is further decreased.

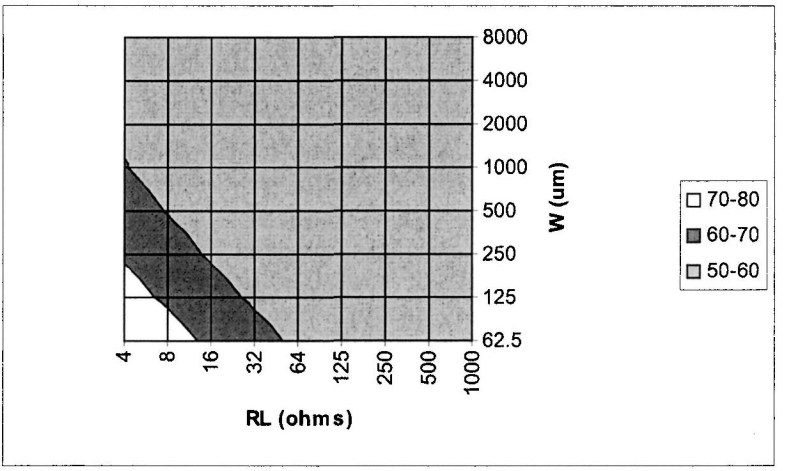


Figure 5-19. DC PSR of a typical source follower voltage regulator

The apparent discrepancy between the simulated high-frequency response of Figure 5-18 and what actually occurs can be explained by the assumption taken in the beginning of this section. It was assumed that an ideal bias voltage is applied to the gate of the source follower device. As frequency increases, impedance of the C_{gs} parasitic capacitance of the device decreases and more amplitude variation appears on the gate. However, since an ideal input was assumed to the gate voltage, no variation appears at the output.

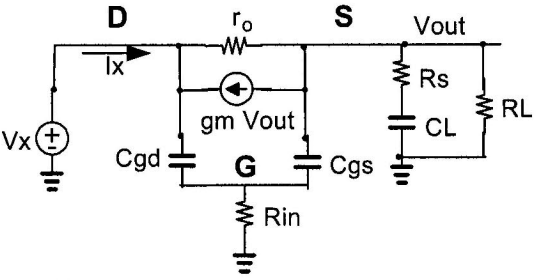


Figure 5-20. Realistic AC small-signal model of source follower with typical load

Figure 5-20 shows a more realistic AC small-signal model of the NFET source follower with a typical load. The resistor R_{IN} models the finite input impedance of the input bias circuit. The addition of resistor R_{IN} causes the system to be a three pole and three zero system. If the bias circuit has low output impedance (low R_{IN}), the zeros and poles introduced by C_{gs} and C_{gd} should not affect the PSR for the frequency range of interest. Figure 5-21 shows the frequency response of the NFET source follower with low input impedance on the gate terminal. As the figure shows, the exhibits a two zero and two pole system, since C_{gd} was small. This curve also shows two important characteristics. First, the PSR degrades for mid-frequencies. Secondly, the PSR flattens for higher frequencies. Figure 5-22 shows the frequency response of the source follower with high input impedance. As the figure shows an extra pole and zero were added to the system.

The disadvantage of a source follower voltage regulator is that, unlike a linear regulator, the internal supply can vary significantly with process and loading conditions. The V_{GS} of the source follower device increases with the square root of the current draw from the supply. Another disadvantage of source follower regulators is that the regulated supply is at most one V_{GS} lower than the supply. In many cases this translates to 0.5V to 1.0V of headroom loss, a significant figure for low-voltage PLLs [12-18].

One method to recover the lost headroom is to use charge pumps [50]. A charge pump is a circuit that is capable of producing a stable output voltage greater than the supply voltage. A sample charge pump is shown in Figure 5-23. This particular charge pump is known as a Dickson charge pump [50]. This is not to be confused with charge pumps used in PLLs. As the figure below shows, the charge pump is a clocked system that is capable of producing an output voltage higher than the supply voltage. The disadvantage of a charge pump is its inability source large current.

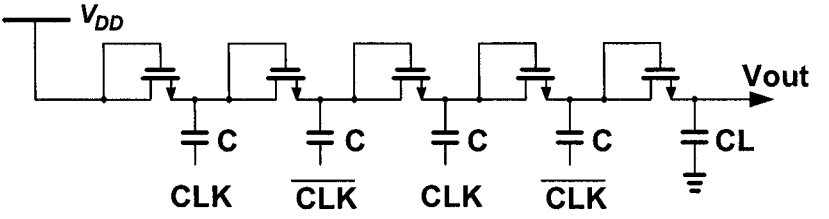


Figure 5-21. A Dickson charge pump schematic

Each stage of the Dickson charge pump consists of a diode capacitor pair. Current from the clock buffer is pushed into the internal nodes forcing the voltage to be higher by an amount equal to

$$\Delta V = V_{dd} \cdot \frac{C}{C + C_s} \tag{5.22}$$

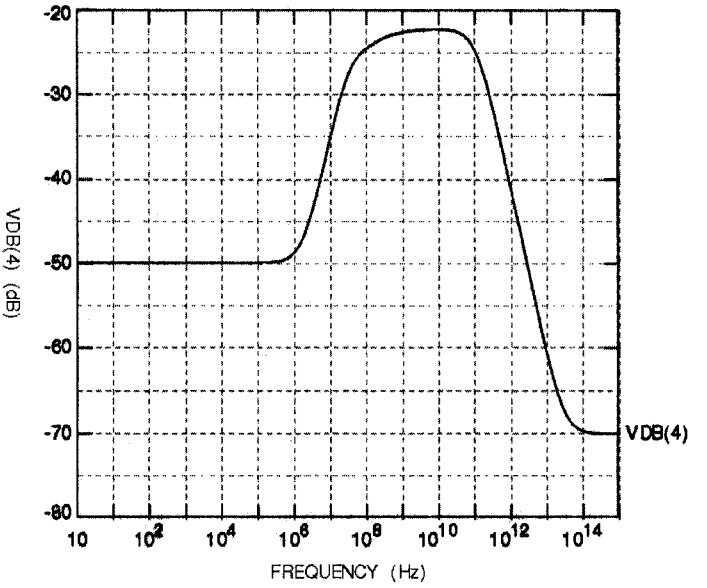


Figure 5-22. Frequency response of SF with typical load and low input bias impedance

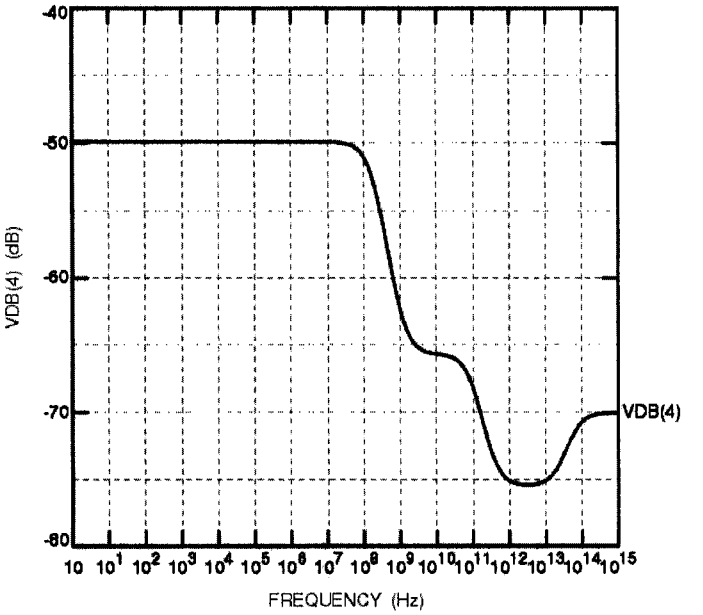


Figure 5-23. Frequency response of SF with typical load and high input bias impedance

where C is a capacitor sized large enough to maximize the signal amplitude feed-through from the clock buffers to the charge pump internal nodes, C_s is the parasitic capacitance at each internal charge pump node, and V_{dd} is assumed to be the single-ended swing of the clock buffer. Since the NFET devices are connected in diode configuration, the efficiency of the charge pump goes down by a $V_{TN}(V_x)$ at each stage. A discussion on low-voltage high-efficiency charge pumps can be found in [51-53]. It is important to node that body effect increases the threshold voltage, V_{TN}, of the device by an amount proportional to the internal voltage V_x at each stage. The output voltage of a Dickson charge pump can be given as

$$V_{out} = n \cdot \left(V_{dd} \cdot \frac{C}{C + C_s} - V_{TN}(V_x) \right) \tag{5.23}$$

where n is the number of pairs of diode connected NFETs and feed-through capacitors C. Using a supply voltage of 2.5V, and a nominal threshold voltage of 0.5V, the four stage Dickson charge pump in Figure 5-23 was simulated and the transient output waveform is shown in Figure 5-24. Neglecting the body effect, the expected output is 8V. The output waveform of the figure is reasonably close to this value. A 50MHz non-overlapping 50% duty cycle clock signal was used in this simulation.

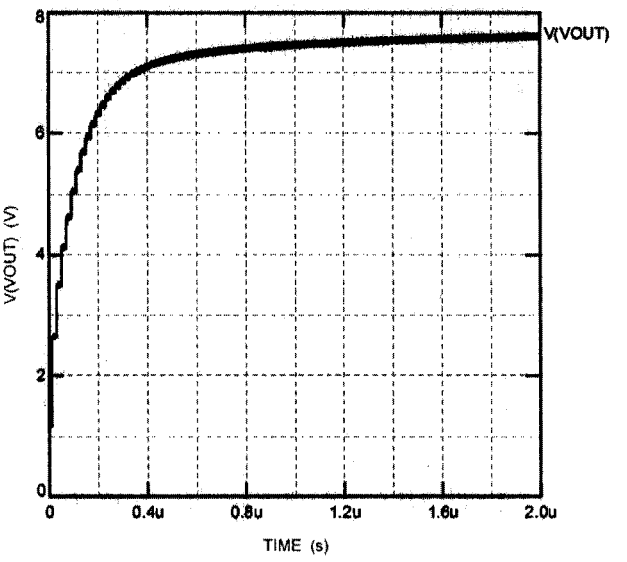


Figure 5-24. Output transient waveform of a four-stage Dickson charge pump

One way to use a charge pump in a source follower regulator is shown in Figure 5-25 [22]. The charge pump is applied to the gate terminal of the

source follower NFET device. Since the charge pump is applied to the bias voltage for the source follower device, it does not need to source large current. Using this technique, a low supply voltage can be used while still maintaining reasonable headroom to operate the ICO reliably.

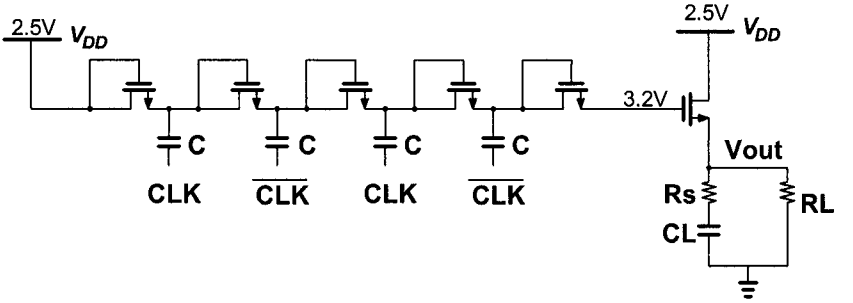


Figure 5-25. Source follower voltage regulator using charge pumps for low-voltage operation

The disadvantage of this technique is that it relies on a clock signal to produce the required high voltages. As the waveform in Figure 5-24 suggests, the output of the charge pump contains ripples with a period equal to the driving clock signal. In most cases, the clock signal is higher than the closed loop bandwidth of the PLL. Since the current produced by the charge pump is injected directly into the ICO, charge pump clock spurs are not filtered by the PLL closed loop response.

One method of reducing the magnitude of the spurs generated by the charge pump is to add a larger capacitive load to the ICO regulated power supply. This, however, would increase the settling time of the charge pump. The settling time, in turn, can be enhanced by increasing the frequency of the input clock of the charge pump.

5.3.2.3 Hybrid Voltage Regulation Approach

The above techniques can be used together to maximize results. For example, in [6] a source follower has been used along with a linear regulator as shown in Figure 5-26. Simulations of this topology indicate that a PSR exceeding 50dB are achievable. In the first stage, a source follower is used to regulate the power supply to an intermediate voltage. Decoupling capacitors to ground and resistors to supply are used on the input bias voltage to stabilize the gate terminal voltage of the NFET device with respect to the supply voltage. Since the potential of the gate voltage is the supply voltage, the output source voltage is maximized. The second stage consists of a linear regulator with a PFET pass gate operated in the linear region. The output voltage is well regulated within the voltage regulators loop bandwidth to a voltage proportional to a reference voltage. This

reference voltage can be supplied from a bandgap reference generator. The proportionality factor is controlled by the switches on the R1 resistor.

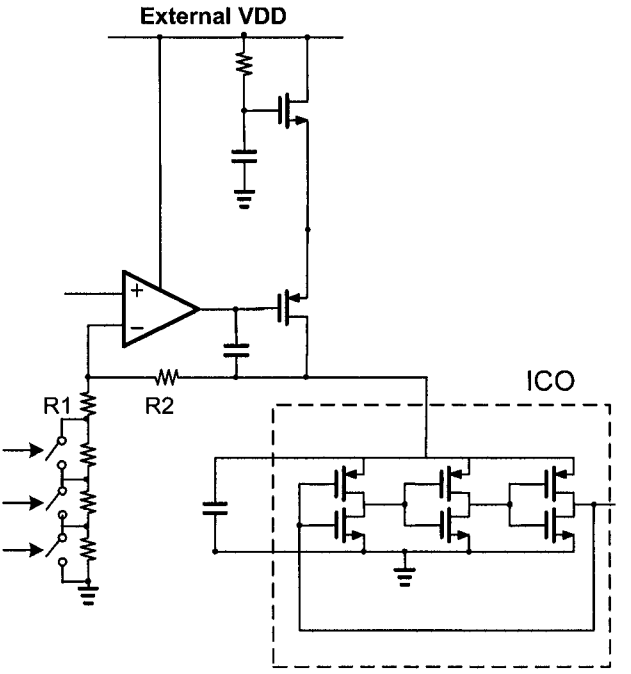


Figure 5-26. Hybrid source-follower linear regulator approach

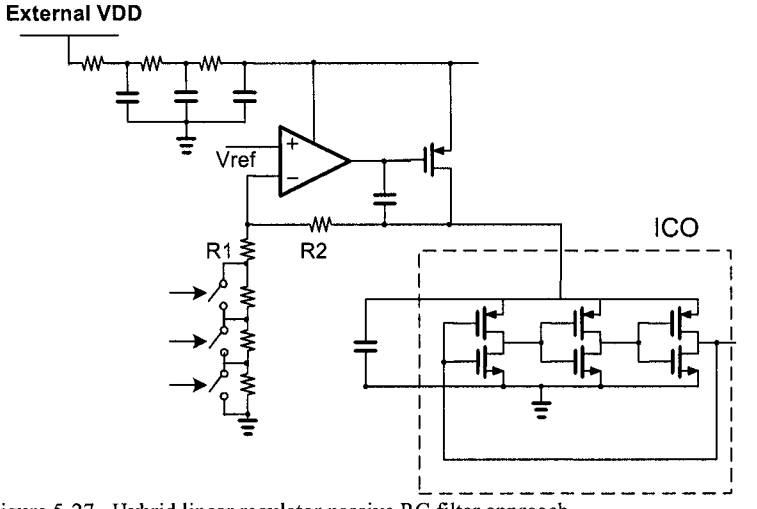


Figure 5-27. Hybrid linear regulator passive RC filter approach

Another hybrid approach uses a linear regulator with a passive RC filter [9], as shown in Figure 5-27. As mentioned earlier, one of the disadvantages of a linear regulator approach is that the PSR degrades rapidly beyond the regulator's closed loop bandwidth. As demonstrated in chapter 4, noise affects the ICO only near dc, ICO operating frequency, and multiples of the ICO operating frequency. Therefore, the passive RC filter only requires good attenuation starting at the ICO operating frequency. One disadvantage of an RC filter is the IR voltage drop required across the filter's resistor, which reduces the supply voltage level available for the ICO.

5.3.3 Performance Comparison

Each of the above mentioned techniques has its own merits and downsides. In most PLLs for SoC processors, the major contributor of jitter is the supply voltage. The immunity of the PLL to supply voltage noise is important. One way to measure them, as mentioned in chapter 4, is to inject the PLL supply voltage with a periodic step voltage and measure the jitter resulting for the supply voltage step. The rating is usually expressed in units of picoseconds of jitter per volt of unit step size, or ps/V. The voltage step is usually a square waveform with a frequency chosen to be less than or equal to 1MHz (where power supply rejection is good!).

Table 5-1. Comparison of various PLL supply noise rejection techniques

Refs	Technology	Power supply noise immunity (ps/V)	Techniques used
[10]	0.35um CMOS	80 ps/V	Linear voltage regulator with PMOS cascades, single-ended ICO
[20]	0.35um CMOS	385 ps/V	Maneatis loads, differential ICO (no power supply regulation)
[7]	0.18um CMOS	45 ps/V	High gain linear regulator and source follower, single-ended ICO
[22]	0.35um CMOS	70 ps/V	Boosted source follower controlled by linear regulator and RC filtering on supply, single-ended ICO
[11]	0.6um CMOS	200 ps/V	Boosted source follower and single-ended ICO
[9]	0.8um CMOS	200 ps/V	Linear regulator with RC filtering on supply, single-ended Maneatis loads
[21]	0.18um CMOS	250 ps/V	Linear regulator with PMOS cascode, single-ended ICO

Table 5-1 above shows the performance of each technique using the above mentioned figure-of-merit (ps/V). The figures were obtained from recently published literature. As it can be seen there is as much as a 10x improvement in power supply noise immunity in PLLs with active power supply regulation in comparison to a differential ICO based PLL with no voltage regulation.

5.4 Adaptive PLL Architectures

The previous section demonstrated PLL architectures that are capable of reducing jitter due to power supply noise. In general, power supply noise is a time-varying random function with varying power distribution. Although power supply regulation minimizes this contribution to jitter, it may be costly in terms of area and voltage headroom.

Power supply noise and intrinsic ICO noise are sited as the main contributors to jitters in phase-locked loops for SoC applications [8,19]. Although this is true, optimizing the loop bandwidth for both may require a balance of jitter contribution due to various noise sources. In other cases, it may be difficult to find a closed loop bandwidth that meets both the lock time and low-jitter performance requirements. To further agitate matters, process, voltage, and temperature variations may alter the optimal closed loop bandwidth may vary by as much as ±40%. In this section, a variety of adaptive bandwidth techniques are explored.

5.4.1 Two-Mode Bandwidth Architecture

A two-mode bandwidth architecture is used in situations where the jitter and lock time requirements produce contradictory optimal design points. A two-mode bandwidth PLL architecture is shown in Figure 5-28 [29]. As the figure shows, there are two charge pumps and a switch on the loop filter. During frequency acquisition, the charge pump with the larger current is used and the switch bypasses the resistor in the loop filter. This causes the loop filter capacitor to be charged directly and quickly to its optimal value. When near lock condition, the loop filter switch is open and the second charge pump (with lower current) is used. This helps to limit the loop bandwidth during steady-state operation.

This type of PLL is used in applications where the noise contribution from the TCXO buffer, charge pump, or noise injected into the loop filter is significant and when input reference frequency to closed loop bandwidth ratio is more than 100. The disadvantage of this technique is that when the PLL mode is altered from one mode to the other, a sizable glitch may be

created on the loop filter. This causes the PLL to re-lock and thus limits the maximum possible reduction in lock time.

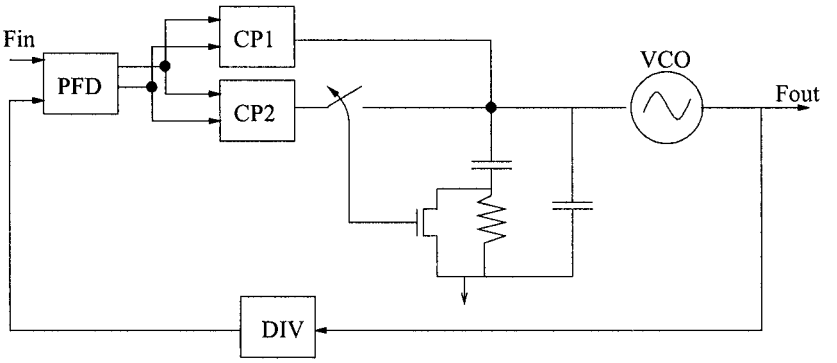


Figure 5-28. Two-mode bandwidth PLL architecture

5.4.2 Process Insensitive Design

Modern PLLs for SoCs are fully integrated. Furthermore, to reduce costs, loop filter capacitors and resistors are usually implemented from MOSFET transistors using the gate capacitance and channel resistance, respectively. For these two reasons, process, temperature, and voltage variations can change the loop bandwidth by as much as ±40%. This translates to large variation in lock time and jitter performance. One method of reducing this sensitivity is to design a process-independent PLL based on self-biased techniques [30].

The main objective of this technique is to control the ratio of the output frequency to the PLL's closed loop response under different operating conditions constant. This is equivalent to controlling both the damping factor as well as the maintaining a constant ratio of output frequency to natural frequency (for a fixed N).

To understand how this method works, recall that

$$\omega_n = \sqrt{\frac{K_o I_p}{2\pi NC}} \tag{5.25}$$

and

$$\zeta = \frac{\omega_n RC}{2} = \frac{1}{2} \cdot \sqrt{\frac{I_p K_o R^2 C}{N}} \tag{5.26}$$

Note that I_p, K_o, R and C all vary with process. Process-independent PLL based on self-biased techniques attempt to reduce this variation by fixing the charge pump current to be a multiple of the bias current used in the ICO and the resistor to be inversely proportional to the square root of the ICO bias current, I_D. From (5.25) and (5.26) it can be seen that the damping factor would be constant and the natural frequency will be proportional to the square root of I_D.

Consider an ICO that uses Maneatis loads as delay cells, as shown earlier in chapter 4. The output swing is inversely proportional to the g_m of the Maneatis loads (also known as symmetric loads). Since the output frequency is proportional to the effect 1/RC of each ICO stage, the output frequency is proportional to the square root of I_D, or linearly proportional to the VCO control voltage. This indicates that, to the first order, the VCO gain, K_o, is constant with respect to I_D. Under these assumptions, it can be seen that the output frequency as well as the natural frequency are proportional to the square root of I_D, or constant with respect to each other.

The PLL architecture shown in Figure 5-29 can be used to implement a self-baised PLL. Note that in order to implement a resistor that is proportional to the inverse of the square root of I_D, the series RC loop filter had to be split into two branches. The lower branch produces the required ΔV_{FILT} due to the capacitor and the upper branch produces the required ΔV_{FILT} due to the resistor branch. Note that the resistor in this case is the $1/g_m$ of the replica bias branch of the ICO (as stated earlier, the g_m of the symmetric load is inversely proportional to the square root of I_D).

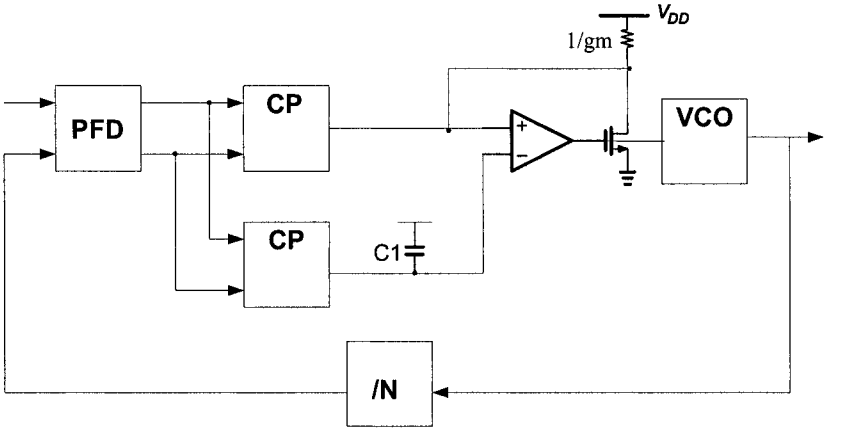


Figure 5-29. Self-biased PLL Architecture

From Figure 5-29, one can see that the damping factor can now be expressed as

$$\zeta = \frac{\omega_n (1/g_m) C}{2} = \frac{1}{4} \cdot \sqrt{\frac{x \cdot C_1}{N \cdot C_B}} \tag{5.27}$$

where x is a proportionality factor linearly relating the I_p to I_D (usually 2), and C_B is the total switched capacitance in the ICO, and C_1 is the loop filter capacitance. The ratio of the natural frequency to the output frequency can be expressed as

$$\frac{\omega_n}{\omega_{out}} = \frac{x \cdot N}{2\pi} \sqrt{\frac{C_B}{C_1}} \tag{5.28}$$

Both equations (5.27) and (5.28) show a dependence on the ratio of two capacitors, which can be matched quite well. Other than this capacitor ratio, the PLL response is independent of process variation as desired.

There are two main disadvantages to this technique. First, the charge pump current is produced by a replica bias stage, connected in feedback configuration. This method of generating a current for the charge pump is noisy and may be prone to stability issues. The second disadvantage is that the self-biased PLL is immune to process, temperature, and voltage variations only. It however, does vary with feedback division ratio. This means that performance will vary with output frequency (loop bandwidth variation with f_{out}). The main reason for this is that the natural frequency was kept constant with respect to the output frequency, instead of the input frequency. A method of circumventing this second disadvantage is introduced in section 5.5

5.4.3 Analog Loop Bandwidth Controller

Another definition of "adaptive PLL" architecture is to dynamically vary the loop bandwidth of the PLL to achieve optimal jitter performance. The two steps in the transient behavior in any PLL is frequency and phase locking followed by frequency and phase tracking. The first step requires a large loop bandwidth in order to achieve fast locking. Fast locking is becoming more important in SoC processors due to the proliferation of heterogeneous clock domains on-chip. The second step involves frequency and phase tracking, requiring an "optimal" loop bandwidth that minimizing jitter contribution of VCO versus other components in the PLL [23-27]. In section 5.4.1, a two-mode control technique was introduced that offers a compromise between these two requirements. In this section, an alternate solution is presented.

One method of achieving fast locking during PLL locking is to modify the loop filter updating technique. The loop filter is updated once every reference period. If the reference period is fixed (which is the case in most

practical applications), the stepping the loop filter voltage by an optimal amount is important. Such optimal gear-shifting shifting algorithms were studied in [47-48]. Such techniques rely on recursive least squares (RLS) adaptive algorithms or Kalman filtering techniques, which are computationally intensive.

An alternative method is to realize a simplified RLS algorithm in analog form [25]. The constant gain term in the PLL's open loop response is updated by the following equation:

$$K_{loop}(n+1) = \lambda \cdot K_{loop}(n) + \alpha \cdot \left| \theta_{input}(n) - \theta_{output}(n) \right| \tag{5.29}$$

where λ is a "forgetting factor" (somewhat less than unity) and α is a gain factor. In control theory terminology, equation (5.29) is the estimator of the adaptive algorithm. The estimator is implemented in analog form as shown in Figure 5-30.

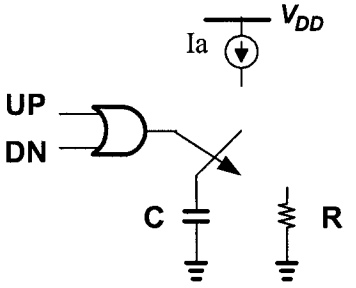


Figure 5-30. Analog implementation of estimator

Whenever UP or DN is on, the capacitor is charge linearly by current source I_a. When UP and DN are off, the capacitor is discharged through resistor R exponentially at a rate proportional to RC. The linear rate (controlled by the I_a/C ratio is equal to α, and the exponential decay rate, RC, is equal to λ. The resulting voltage across the capacitor is then used as a bias voltage to control the charge pump current adaptively. The lowest charge pump current needed to minimize the phase error (time when output of OR gate is high) is then automatically adjusted. The overall PLL architecture is shown in Figure 5-31.

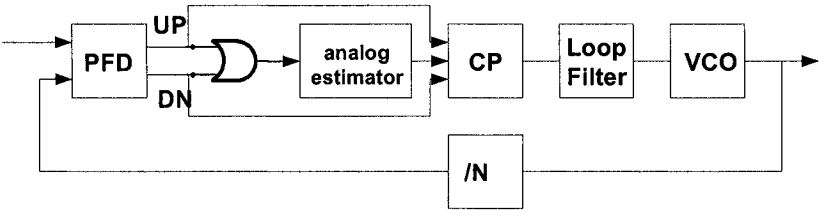


Figure 5-31. Adaptive loop bandwidth PLL Architecture

5.5 Resistorless Loop Filter PLLs

5.5.1 Basic Concept

One of the main contributors of reference spurs, or deterministic jitter, in charge pump PLLs is the presence of the resistor in the loop filter. The resistor is required in order to stabilize the closed loop response of the PLL; however, as explained in earlier chapters, the loop filter voltage ripple due to the filter's resistor causes objectionable reference spurs. This reference spur can be reduced by adding a second pole to the loop filter, attenuating the ripple.

Alternatively, the resistor can be eliminated altogether. To understand how this can be accomplished, the loop filter components are split into integral (associated with the main capacitor) and proportional (associated with the main resistor) terms. The integral component sums the phase errors, which have been converted to current pulses, and represents that as a voltage. The proportional term has no such memory and directly translates the phase error pulse into a voltage pulse. The resistor can be eliminated altogether by adding this current pulse into the ICO directly (with some scaling factor). The self-biased PLL (Figure 5-29) is one example where the resistor loop filter has been eliminated altogether. This has been accomplished by adding a feedforward current branch directly into the ICO. Another promising solution is described below.

5.5.2 Sample-Reset Loop Filter

The sample-reset loop filter architecture [31,33] is a resistor-less class of PLL architectures that attempts to reduce the PLL's output reference spur by reducing the magnitude of the current spike due to the proportional term. The unique solution of the sample-reset loop filter architecture is that the current pulse width is fixed and occupies most of the reference period. The amplitude of the current pulse is what determines the phase error injected into the loop filter. This technique results in dramatic reduction in reference spur and also helps to boost the PLL's phase margin [33].

The sample-reset loop filter architecture is shown in Figure 5-32. The integral part of the loop filter consists of the main charge pump, C_1, and g_{m1} transconductance stage. The phase errors from the main charge pump are accumulated by capacitor C_1 and converted into a current by g_{m1}. This current is then used to modify the current in the ICO during the update cycle. The proportional part of the loop filter consists of auxiliary charge pump, C_2, and g_{m2}. The charge pump is charge capacitor C_2 during the evaluation

period. During the update period, the voltage across the capacitor is converted into a current by g_{m2} and fed into the ICO. After the update period, the voltage on capacitor C_2 is reset in order to avoid adding an extra pole. Techniques similar to the ones presented in section 5.4.2 are also used to ensure process, voltage, and temperature independent performance. Another advantage of this technique is that loop dynamics do not vary with N.

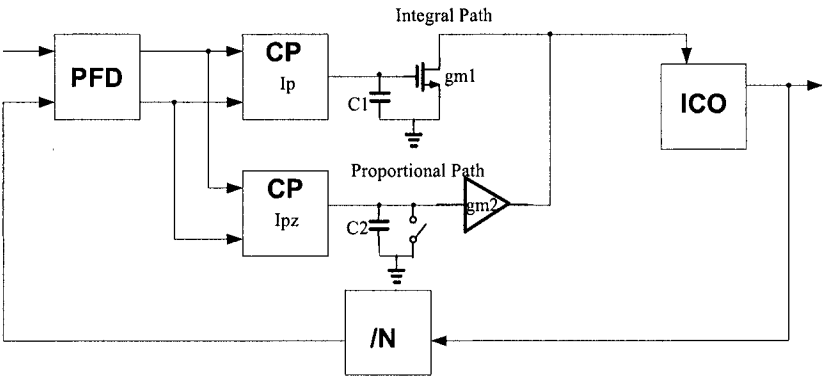


Figure 5-32. Sample-reset loop filter architecture

5.6 Delay-Locked Loop Frequency Multipliers

Thus far, all PLLs have been assumed to have a charge-pump based topology. As has been shown in the previous sections, innovations in the loop filter (either resistor-less or fully differential) have been shown to have certain advantages. In this section, a different paradigm in frequency generation is shown that is based on a *delay-locked loop* (DLL) [34-46]. It is shown that this architecture has strong advantages, but also has certain disadvantages that prevent their widespread adoption.

5.6.1 DLL Operating Principle

A diagram of a basic DLL is shown in Figure 5-33. As the figure shows, a DLL consists of a PFD, charge pump, loop filter, and a voltage-controlled delay line (VCDL). The main difference between a PLL and a DLL is that the VCO is substituted by a VCDL. No frequency multiplication occurs in a DLL.

A voltage controlled delay line (VCDL) consists of buffer or inverter stages cascaded together, as shown in Figure 5-34. Each buffer stage may

consist of a simple inverter or a symmetric load, just as in the case of a VCO. As figure 5-33 indicates the input to the VCDL is the input reference clock. The PFD compares this same input reference clock with a delayed version of the clock (output of the VCDL).

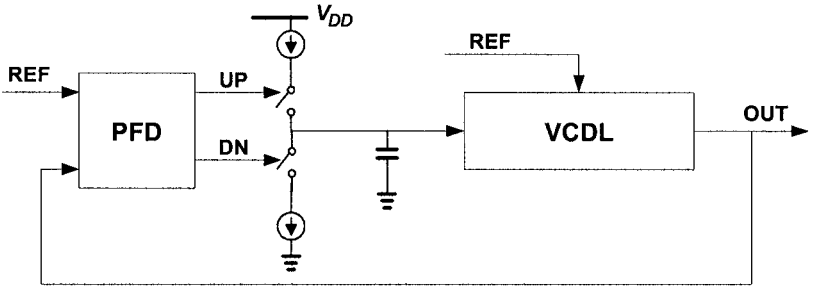


Figure 5-33. Diagram of a basic DLL

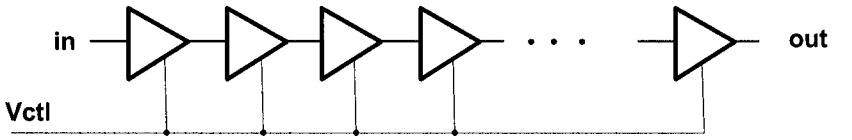


Figure 5-34. Typical voltage-controlled delay line (VCDL)

This behavior of the DLL causes the total delay in the buffer chain to be equal to one reference period. Depending on the gain in the DLL, the DLL may also false lock to a multiple of the input period or to attempt to lock to zero time. This "false locking" can be prevented by either using a pre-tuning algorithm that adjusts the delay in the VCDL to be close to the input reference period or by limiting the gain of the VCDL [35,38,47]. When the DLL is locked, each buffer in the VCDL has a delay of T_{ref}/M, where T_{ref} is the input reference period and M is the number of buffer stages in the VCDL.

Another interesting feature of the DLL is that the loop filter only contains one capacitor. Unlike a PLL, a DLL only corrects for phase error, thus obviating the need for a second pole. Since the closed loop response of the DLL is a single-pole system, it is unconditionally stable. This means that no resistor is needed in the loop filter for stability.

Using the representation given in figure 5-33, the closed loop response from the input to output of the DLL is given as

$$H_1(s) = \frac{I_p \cdot K_v / C}{s + I_p \cdot K_v / C} \tag{5.30}$$

where I_p is the charge pump current, K_v is the VCDL gain (rad/V) and C is the loop filter capacitance. It is interesting to note that the open loop gain is also the closed loop bandwidth of the system (as should be the case of a first order system). Noise injected from the input of the DLL, it would have the same transfer function as equation (5.30). This noise would be low-pass filtered since equation (5.30) indicates a low-pass transfer function. When noise is injected into the loop filter, the transfer function is given as

$$H_2(s) = \frac{K_v / C}{s + I_p \cdot K_v / C} \tag{5.31}$$

This also indicates a low-pass characteristic. As in the case of a PLL, one method of reducing this noise contribution is to increase the charge pump current. Noise injected by the VCDL can be output referred and the DLL noise shaping transfer function would be

$$H_3(s) = \frac{s}{s + I_p \cdot K_v / C} \tag{5.32}$$

This indicates a zero at DC and a pole at the open loop gain. This is a high-pass filter. Therefore, as in the case of the PLL, the VCDL noise is high-pass filtered by the DLL closed loop response. The difference is that the VCDL noise contribution is attenuated only by 20dB/decade within the closed loop bandwidth.

5.6.2 DLL as a Frequency Synthesizer

Although no frequency multiplication occurs in a DLL, the DLL structure can be used in a frequency synthesizer. The basic premise behind this technique is to combine the various phases in the VCDL in such a way that frequency multiplication occurs. This frequency multiplication can occur in open loop (outside feedback path of the DLL) or in closed loop (inside feedback path of the DLL), as shown in Figure 5-35 [34].

Open loop configuration is usually preferred since it does not require a frequency counter and hence consumes less area and power. Although the feedback configuration closely resembles a PLL, it does not behave the same as a PLL since there is no accumulation of jitter in the VCDL (no pole is introduced by the combination of the VCDL and phase selector tree).

In essence frequency multiplication is constructed by having a phase selection tree that performs frequency multiplication. For example, if multiply-by-2 is required, the 0° and 90° phases are chosen and XOR'ed together by the phase selection tree. The XOR operation is effectively multiplication. Multiplying these two signals gives

$$y(t) = \cos(\omega_c t) \cdot \cos\left(\omega_c t + \frac{\pi}{2}\right) = -\frac{1}{2}\sin(2\omega_c t) \tag{5.33}$$

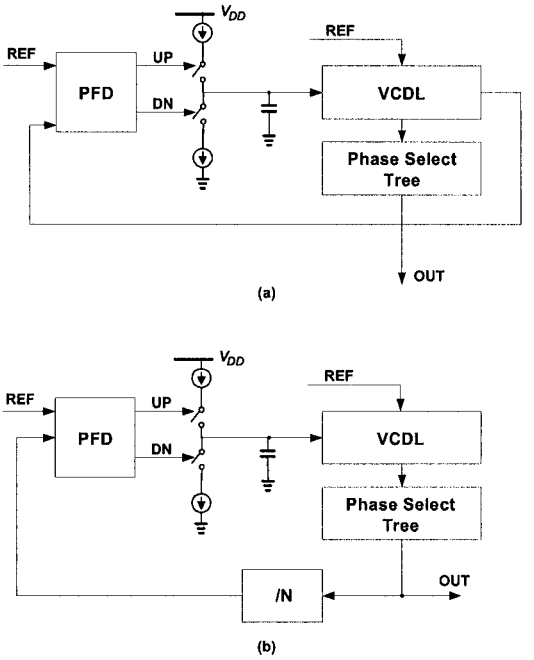


Figure 5-35. Frequency multiplication in a DLL in (a) open loop or (b) closed loop

Therefore, the output waveform has double the frequency of the two inputs. The ½ term can be neglected since the digital logic also acts as an amplifier and voltage limiter fixing the output swing to the power supply.

Another method is based on edge detection and collection [44]. A wide fan-in edge detection logic gate produces a pulse on any positive edge trigger of any of its inputs. Control bits can then individually enable or disable any of the inputs to the wide fain-in edge detection logic gate. The output of the gate is fed into a toggle flip-flop which produces alternating edge triggers (positive then negative) on every positive edge of the logic gate output.

One disadvantage of this configuration is limited frequency multiplication range. If 50% duty cycle output is desired, then it is not possible to generate a continuous range of frequency multiplication ratios. Furthermore, the maximum frequency multiplication ratio is limited to half the number of delay stages in the VCDL.

5.6.3 Jitter Analysis

One important performance metric of the DLL is its jitter performance. As in the case of the PLL, the VCDL generates more noise translating into jitter than any other circuit block. The VCDL can also be made more immune to external noise using the voltage regulation techniques discussed earlier in section 5.3.

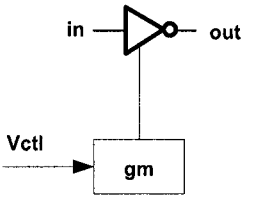


Figure 5-36. Generic voltage controlled delay cell

For analysis purposes, a generic voltage-controlled delay cell shown in Figure 5-36 is assumed. The inverter can be a single-ended inverter or a differential pair. Jitter in the VCDL is assumed to be the total contribution of two sources of noise: noise due to the gm stage in the delay cell and noise in the inverter switching core itself. Assuming a purely capacitive load on each delay cell, the mean square of the total jitter in the VCDL is given as

$$\Delta t_j^2 = 2N \cdot \left(\frac{C_T}{V_{CTL} \cdot g_m} \right)^2 \cdot \Delta v_n^2 \tag{5.34}$$

where C_T is the total switched capacitance in the VCDL, V_{CTL} is the loop filter voltage (control voltage of the VCDL), N is the maximum multiplication ratio, and g_m is the transconductance of the delay cell's gm stage. It is assumed that there are 2·N delay cells in the VCDL. One major source of jitter in DLLs is static mismatch between delay cells caused by process variation. Taking delay mismatch between the VCDL's delay cells into account, the root mean square of the jitter can be given as

$$\Delta t_j = \left(\sum_{i=1}^{2N} \Delta t_{j,mismatch}(i) \right) + \sqrt{2N} \cdot \left(\frac{C_T}{V_{CTL} \cdot g_m} \right) \cdot \Delta v_n \tag{5.35}$$

The phase selection tree would also add jitter to equation (5.35). The jitter added by mismatch, however, is deterministic jitter. When the clock generator is used as a sampling clock for an analog block, deterministic jitter may be more objectionable than random jitter, as is explained in chapter 7. One important result of equation (5.34) and (5.35) is that the jitter increases with desired multiplication ratio. This degradation in jitter performance is not present in PLLs. Therefore, it is expected that for large multiplication

ratios, DLL-based frequency synthesizers can in fact exhibit more jitter than their PLL counterparts.

In order to determine the best frequency synthesis approach, the intrinsic jitter of PLLs and DLLs must be compared. In PLLs, the voltage-controlled oscillator (VCO) accumulates the jitter indefinitely. In other words, any voltage perturbation due to noise is mapped into phase perturbation according to the oscillator's impulse sensitivity function (see chapter 4). This phase shift can only be corrected by the closed loop corrective action of the PLL. The PLL, however, has finite bandwidth and can only correct phase perturbations that occur slower than its loop bandwidth. Therefore, a jitter accumulation factor can be defined and shown to be [28]

$$\alpha = \frac{1}{\sqrt{2I_p \cdot K_v \cdot R \cdot T_{ref}}} \approx \frac{1}{\sqrt{f_{BW} / F_{ref}}} \tag{5.36}$$

where I_p is the charge pump current, K_v is the VCO gain, R is the loop filter resistor value, T_{ref} is the comparison period, f_{BW} is the PLL's closed loop bandwidth and F_{ref} is the comparison frequency. For clock generating PLLs, closed loop bandwidths are usually large and f_{BW}/F_{REF} ratios can reach as high as 0.1. This would give an accumulation factor of approximately 3.

DLLs on the other hand have no accumulation of jitter. However, since the required number of delay elements in a DLL is 2N and a VCO only requires 3 stages, the *effective* accumulation factor in the DLL is 2N/3. If the PLL has an accumulation factor of $\alpha \approx 3$, then the number of delay cells in a DLL cannot be greater than 4 for the DLL-based frequency synthesizer to have less jitter than its PLL counterpart. This limits the multiplication factor to two!

In general, depending on the number of stages in the DLL (dictated by the required multiplication ratio), delay cell mismatches, and jitter in the phase selection tree, the total jitter due to the DLL may be higher or lower. For smaller PLL equivalent loop bandwidths and lower division ratios, DLL-based frequency synthesizers become more attractive. This analysis also assumes noise in the VCO and VCDL is white and uncorrelated, which are not always the case.

5.6.4 Hybrid PLL/DLL Approach

The frequency multiplication difficulties of DLL-based frequency synthesizers are not without remedies. One elegant solution is to use a hybrid PLL/DLL approach [37,42,43]. Figure 5-37 shows one architectural implementation of the hybrid PLL/DLL approach [42].

This architecture first operates by inserting a REF clock edge through the 2-to-1 multiplexer into the three stage VCDL. After the positive edge trigger

of the REF signal, the select logic block sets the multiplexer to configure the VCDL as a ring oscillator. In this configuration, jitter accumulates due to the feedback configuration of the ring oscillator. The div-by-N block divides down the ring oscillator's frequency by a factor of N, where N is the desired multiplication ratio. When the frequency divider finishes counting, the multiplexer opens the ring oscillator loop allowing the REF clock signal into the VCDL. The PFD is also enabled in this mode allowing the VCDL bias voltage to be set closer to the correct value for lock condition.

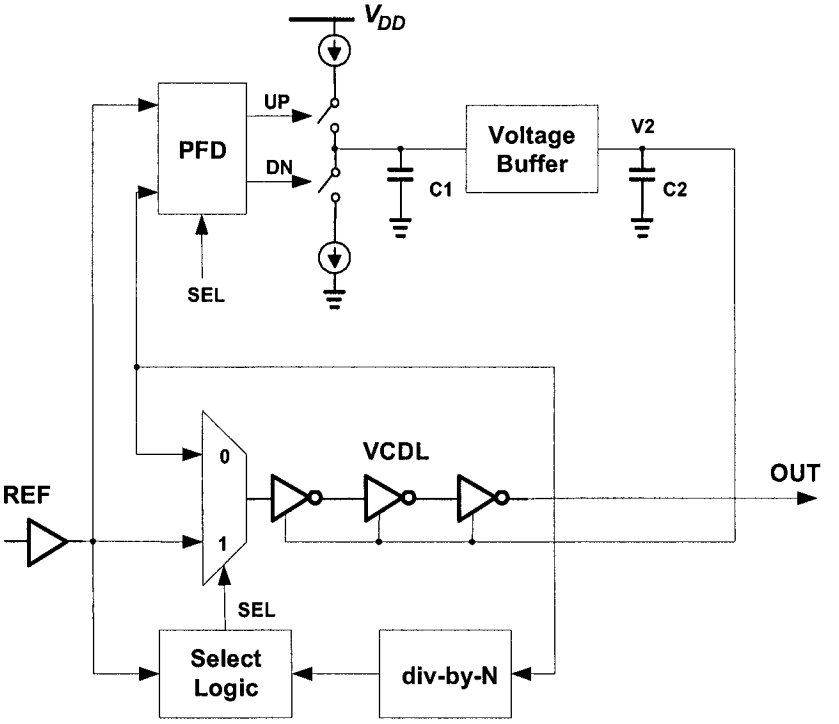


Figure 5-37. PLL/DLL hybrid-based frequency synthesis approach

Opening the loop periodically at the reference frequency rate allows the jitter in the VCDL to be reset. In this way, jitter is not allowed to accumulate for more than one reference period. In modern SoC processors, the reference crystal can be as fast as 20MHz, corresponding to a period of 50ns. For frequencies as high as 500MHz, this corresponds to only 25 VCO cycles.

Variable frequency division is also achieved. As in the case of a PLL, the frequency division ratio is set by controlling the feedback division ratio, N. The effective number of stages of the DLL is three. This means that the

jitter of this architecture always outperforms that of a conventional PLL by a factor equal to the accumulation factor α and that of a pure DLL-based frequency synthesizer by a factor equal to N/3.

5.7 Summary

In this chapter, a multitude of advanced PLL architectures were reviewed and analyzed. The differential PLL architecture was shown to have certain advantages (especially area reduction of the loop filter), but also strong disadvantages. It was shown that the reference spurs of the differential PLL architecture is always worse than a single-ended PLL architecture.

A large section of this chapter was then devoted to the study of voltage regulator design of PLLs. Voltage regulator stabilize the input power supply to the PLL and remove any unwanted tones and power supply noise. Various techniques including PFET cascode structures, enhanced cascode structures, linear power supply regulation, source followers, and hybrid topologies were analyzed. It was shown that power supply regulation can provide up to 50dB improvement is PSR at low frequencies, which translates to more than 10x reduction in jitter due to power supply.

Other architectures including resistorless and adaptive PLL architecture were analyzed. Special topologies such as self-biased PLLs have the merit of having performance immune to process variation, but have some disadvantages including performance variation with synthesized output frequency. Other adaptive PLL approaches demonstrate how to dynamically vary the bandwidth of the PLL to achieve the minimum possible jitter.

Finally, a different paradigm of frequency synthesis, DLL-based frequency synthesis, has been presented. DLL-based frequency synthesis has the promise of delivering very low jitter in comparison to PLL-based frequency synthesizers by avoiding the need for VCOs, which are replaced by VCDLs. Although VCDLs have no accumulation of jitter found in VCOs, the number of cascaded components in VCDLs is often much larger than a simple 3 stage VCO, thus the total jitter in DLL-based frequency synthesizers may actually be more than PLLs. A promising hybrid PLL/DLL approach has been presented that solves many problems found in DLL-based as well as PLL-based frequency synthesizers.

REFERENCES

[1] I. Novof, et. al., "Fully Integrated CMOS Phase-Locked Loop with 15 to 240MHz Locking Range and ±50 ps Jitter," *IEEE J. of Solid-State Circuits*, vol. 30, no. 11, pp. 1259-1266, November 1995.

[2] K. Lakshmikumar, R. Hadaway, and M. Copeland, "Characterization and Modeling of Mismatches in MOS Transistors for Precision Analog Design," *IEEE J. of Solid-State Circuits*, vol. 21, pp. 1057-1066, Dec. 1986.

[3] V. Manassewitsch, *Frequency Synthesizers, Theory and Design*, Boston: Wiley, 1987.

[4] D. Boerstler, "A Low-Jitter PLL Clock Generator for Microprocessors with Lock Range of 340-612 MHz," *IEEE J. of Solid-State Circuits*, vol. 34, no. 4, pp. 513-519, April 1999.

[5] H. Ahn and D. Allstot, "A Low-Jitter 1.9-V CMOS PLL for UltraSPARC Microprocessor Applications," *IEEE J. of Solid-State Circuits*, vol. 35, no. 3, pp. 450-454, March 2000.

[6] J. Ingino, "A 4-GHz Clock System for a High-Performance System-on-a-Chip Design," *IEEE J. of Solid-State Circuits*, vol. 36, no. 11, pp. 1693-1698, November 2001.

[7] J. Ingino, "A 4GHz 40dB PSRR PLL for an SOC Application," *Int'l Solid-State Circuits Conference*, pp. 392-393,469, 2001.

[8] W. Rhee, "Design of Low-Jitter 1-GHz Phase-Locked Loops for Digital Clock Generation," *Int'l Symposium on Circuits and Systems*, vol. 2, pp. 520-523, 1999.

[9] C. Lee, et. al., "Design of Low Jitter PLL for Clock Generator with Supply Noise Insensitive VCO," *Int'l Symposium on Circuits and Systems*, vol. 1, pp. 233-236, 1998.

[10] V. Kaenel, "A High-Speed, Low-Power Clock Generator for a Microprocessor Application," *IEEE J. of Solid-State Circuits*, vol. 33, no. 11, pp. 1634-1639, Nov. 1998.

[11] S. Lee, et. al., "A 3V 200MHz PLL with a Low-Noise VCO Based on a Power-Efficient Low-Ripple DC-DC Converter," *Int'l Conference on VLSI and CAD*, pp. 346-349, 1999.

[12] V. Kaenel, et. al., "A 320 MHz, 1.5 mW @ 1.35 V CMOS PLL for Microprocessor Clock Generation," vol. 31, no. 11, pp. 1715-1722, Nov. 1996.

[13] V. Kaenel, et. al., "A 320MHz, 1.5mW at 1.35V CMOS PLL for Microprocessor Clock Generation," *Int'l Solid-State Circuits Conference*, pp. 132-133, 1996.

[14] J. Alvarez et. al., "A Wide-Bandwidth Low-Voltage PLL for PowerPC™ Microprocessors," *Symposium on VLSI Circuits*, pp. 37-38, 1994.

[15] J. Montanaro, et. al., "A 160-MHz, 32-b, 0.5-W CMOS RISC Microprocessor," *IEEE J. of Solid-State Circuits*, vol. 31, no. 11, pp. 1703-1714, Nov. 1996.

[16] H. Ahn and D. Allstot, "A Low-Jitter 1.9-V CMOS PLL for UltraSPARC Microprocessor Applications," *IEEE J. of Solid-State Circuits*, vol. 35, no. 3, pp. 450-454, March 2000.

[17] H. Ahn, "A +-25ps 1.9V CMOS PLL for UltraSPARC microprocessor," *IEEE Custom Integrated Circuits Conference*, pp. 303-306, 1999.

[18] J. Alvarez, et. al, "A Wide-Bandwidth Low-Voltage PLL for PowerPC™ Microprocessors," *IEEE J. of Solid-State Circuits*, vol. 30, no. 4, pp. 383-391, April 1995.

[19] F. Anderson, J. Wells, and E. Berta, "The Core Clock System on the Next Generation Itanium™ Microprocessor," *Int'l Solid-State Circuits Conference*, 8.5, 2002.

[20] I. Young, M. Mar, B. Bhushan, "A 0.35um CMOS 3-880MHz PLL N/2 Clock Multiplier and Distribution Network with Low Jitter for Microprocessors," *Int'l Solid-State Circuits Conference*, pp. 330-331, 1997.

[21] V. Kaenel, et. al., "A 600MHz CMOS PLL Microprocessor Clock Generator with a 1.2GHz VCO," *Int'l Solid-State Circuits Conference*, pp. 896-897, 1998.

[22] C. Lee, K. McClellan, and J. Choma, "A Supply-Noise-Insensitive CMOS PLL with a Voltage Regulator Using DC-DC Capacitive Converter," *IEEE J. of Solid-State Circuits*, vol. 36, no. 10, pp. 1453-1463, Oct. 2001.

[23] J. Kim et. al, "Design of CMOS Adaptive-Bandwidth PLL/DLLs: A General Approach," *IEEE Trans. On Circuits and Systems II*, vol. 50, no. 11, pp. 860-869, November 2003.

[24] M. Mansuri, A. Hadiashar, C. Yang, "Methodology for on-Chip Adaptive Jitter Minimization in Phase-Locked Loops," *IEEE Trans. On Circuits and Systems II*, vol. 50, no. 11, pp. 870-878, November 2003.

[25] J. Lee and B. Kim, "A Low-Noise Fast-Lock Phase-Locked Loop with Adaptive Bandwidth Control," *IEEE J. of Solid-State Circuits*, vol. 35, no. 8, pp. 1137-1145, August 2000.

[26] M. Bayer, et. al., "A Low Noise CMOS Frequency Synthesizer with Dynamic Bandwidth Control," *IEEE Custom Integrated Circuits Conference*, pp. 171-174, 1994.

[27] J. Lee and B. Kim, "A 250MHz Low Jitter Adaptive Bandwidth PLL," *Int'l Solid-State Circuits Conference*, pp. 346-347, 1999.

[28] B. Kim, T. Weigandt, P. Gray, "PLL/DLL System Noise Analysis for Low Jitter Clock Synthesizer Design," *IEEE Int'l Symposium on Circuits and Systems*, pp. 31-34, 1994.

[29] H. Sato, K. Kato, and T. Sase, "A Fast Pull-In PLL IC Using Two-Mode Pull-In Technique," *Electronics and Communications in Japan*, Part 2, vol. 75, no. 3, pp. 41-50, 1992.

[30] J. Maneatis, "Low-Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques," *IEEE J. of Solid-State Circuits*, vol. 31, no. 11, pp. 1723-1732, November 1996.

[31] A. Maxim, et. al., "A Low Jitter 125-1250MHz Process Independent 0.18um CMOS PLL Based on a Sample-Reset Loop Filter," *Int'l Solid-State Circuits Conference*, pp. 394-395, 2001.

[32] B. Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw-Hill: Boston, 2001.

[33] A. Maxim, et. al, "Sample-Reset Loop Filter Architecture for Process Independent and Ripple-Pole-Less Low Jitter CMOS Charge-Pump PLLs," *Int'l Symposium on Circuits and Systems*, vol. 4, pp. 766-769, 2001.

[34] D. Birru, "A Novel Delay-Locked Loop Based CMOS Clock Multiplier," *IEE Trans. On Consumer Electronics*, vol. 44, no. 4, pp. 1319-1322, 1998.

[35] D. Foley and M. Flynn, "CMOS DLL Based 2V, 3.2ps Jitter, 1GHz Clock Synthesizer and Temperature Compensated Tunable Oscillator," *IEEE Custom Integrated Circuits Conference,* pp. 371-374, 2000.

[36] O. Susplugas and P. Philippe, "Multimode clock generation using delay-locked loop," *Electronics Letters*, vol. 39, no. 4, pp. 347-349, 20th February 2003.

[37] M. Lee, et. al, "Jitter Transfer Characteristics of Delay-Locked Loops – Theories and Design Techniques," *IEEE J. of Solid-State Circuits*, vol. 38, no. 4, pp. 614-621, April 2003.

[38] D. Foley and M. Flynn, "CMOS DLL-Based 2-V 3.2-ps Jitter 1-GHz Clock Synthesizer and Temperature-Compensated Tunable Oscillator," *IEEE J. of Solid-State Circuits*, vol. 36, no. 3, pp. 417-423, March 2001.

[39] G. Chion and P. Gray, "A 900MHz Local Oscillator using a DLL-Based Frequency Multiplier Technique for PCS Application," *Int'l Solid-State Circuits Conference,* pp. 202-203, 2000.

[40] C. Wang, H. She, and R. Hu, "A 1.2 GHz Programmable DLL-Based Frequency Multiplier for Wireless Applications," *Int'l Conference on Electronics, Circuits, and Systems,*vol. 1, pp. 69-72, 2002.

[41] R. Beek, et. al., "Low-Jitter Clock Multiplication: A Comparison Between PLLs and DLLs," *IEEE Trans. On Circuits and Systems II*, vol. 49, no. 8, pp. 555-566, August 2002.

[42] R. Frajad-Rad, et. al., "A Low-Power Multiplying DLL for Low-Jitter Multigigahertz Clock Generation in Highly Integrated Digital Chips,"

IEEE J. of Solid-State Circuits, vol. 37, no. 12, pp. 1804-1812, December 2002.

[43] J. Begueret, et. al., "Clock Generator using factorial DLL for Video Application," *IEEE Custom Integrated Circuits Conference*, pp. 485-488, 2001.

[44] C. Kim, et. al., "A Low-Power Small-Area ±7.28-ps-Jitter 1-Ghz DLL-Based Clock Generator," *IEEE J. of Solid-State Circuits*, vol. 37, no. 11, pp. 1414-1420, November 2002.

[45] C. Kim, I. Hwang, S. Kang, "Low-Power Small-Area ±7.28-ps-Jitter 1-GHz DLL-Based Clock Generator," *Int'l Solid-State Circuits Conference*, 8.3, 2002.

[46] J. Lee, S. Han, H. Yoo, "A 330MHz Low-Jitter and Fast-Locking Direct Skew Compensation DLL," *Int'l Solid-State Circuits Conference*, pp. 352-353, 2000.

[47] G. Roh, Y. Lee, and B. Kim, "An optimum phase-acquisition technique for charge-pump phase-locked loops," *IEEE Transactions on Circuits and Systems II*, vol. 44, pp. 729-740, Sept. 1997.

[48] J. Lee and B. Kim, "A 250MHz Low Jitter Adaptive Bandwidth PLL," *Int'l Solid-State Circuits Conference,* pp. 346-347, 1999.

[49] D. Johns and K. Martin, *Analog Integrated Circuit Design*, New York: John Wiley & Sons, Inc. 1997.

[50] J. F. Dickson, "On-Chip High Voltage Generation in MNOS Integrated Circuits Using an Improved Voltage Multiplier Technique," *IEEE J. of Solid-State Circuits*, vol. SC-11, no. 3, June 1976, pp. 374-378.

[51] J. Wu, and K. Chang, "MOS Charge Pumps for Low-Voltage Operation," *IEEE J. of Solid-State Circuits*, vol. 33, no. 4, pp. 592-597, 1998.

[52] Y. Moisiadis, I. Bouras, and A. Arapoyanni, "A CMOS charge pump for low voltage operation," *IEEE Int'l Symposium on Circuits and Systems*, vol. 5, pp. 577-580, 2000.

[53] S. Lai and J. Wang, "A High-efficiency CMOS charge pump circuit," *IEEE Int'l Symposium on Circuits and Systems*, vol. 4, pp. 406-409, 2001.

Chapter 6

DIGITAL PLL DESIGN

6.1 Introduction

In the previous chapter, several PLL topologies have been explored in order to enhance performance, decrease power, or decrease area. All topologies, however, assumed the basic structure of a charge pump PLL, where an UP and DOWN current sources control the voltage of a passive loop filter. In this chapter, an alternate PLL topology is explored where the passive loop filter is substituted by an equivalent digital signal processing block. Numerous attempts at this architecture have been made in the past [18-26,29]. In this chapter, the more significant digital PLLs contributions are categorized and summarized.

6.2 Basic Topology

A diagram of a charge-pump based PLL is shown in Figure 6-1 [19, 28]. A PLL consists of five main blocks: a phase-frequency detector (PFD), a charge pump, a loop filter (usually a 2^{nd} order RC filter), a voltage controlled oscillator (VCO), and a frequency divider of ratio N.

One of the most challenging blocks to integrate is the loop filter. For clock generation PLLs, the loop filter can consume as much as 50% of the total area. The capacitors are usually implemented as MOSFET gate capacitors. For large system-on-a-chip (SOC) applications, the PLL shares the same substrate as a large digital block. Much of the noise can couple through the substrate. Special technologies and techniques such as deep trench isolation, high resistivity substrates [3], and large guard rings are usually required for good isolation. As device sizes shrink, the oxide

thickness also shrinks. This is an important factor enabling supply voltage scaling. This is problematic, however, for PLL designs, since the gate leakage increases exponentially with oxide thickness reduction [3]. Gate leakage, in this case, would cause large reference spurs, which increases jitter and ultimately can result in loss of lock. Recent implementations attempted to address this issue, but on the expense of added complexity, larger active current consumption, and larger area [12].

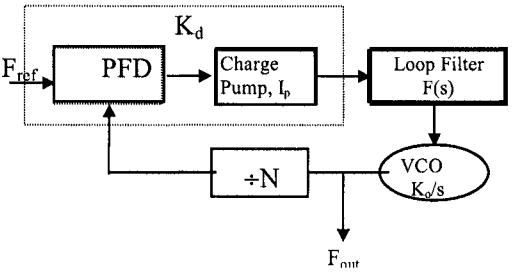


Figure 6-1. A typical charge-pump based PLL

In this chapter, an alternative method for PLL design, which is amenable to SOC mixed-signal design is explored. A PLL with a digital loop filter is explored. The main advantage of the PLL with a digital loop filter is that it avoids all the issues listed above and its performance actually improves with technology (as will be shown later on in this section).

The main challenge in DPLL design is to obtain a wide bandwidth PLL with high frequency resolution. Frequency resolution limitation comes from the fact that the PFD can only quantize the phase error by a high frequency clock, such as the VCO. In one previous DPLL implementation, this was done by effectively creating a frequency locked loop [2]. The DPLL still suffered from coarse granularity of frequency in the PFD, and hence its jitter performance was limited. Moreover, the VCO was required to be reset after each PFD measurement. This places an upper limit of the operating frequency of the PLL.

In [3], another solution the PFD granularity issue was presented. In this scheme, the loop filter is updated once every *m* cycles, where *m* is an integer multiple of the reference frequency. For higher accuracy, more PFD measurements are taken before updating the loop filter. This places a stringent upper limitation on the loop bandwidth.

In this section, an alternate method is used to obtain high frequency accuracy while maintaining a high loop bandwidth is presented [6]. Figure 6-2 shows a block diagram of the digital PLL (DPLL). The PFD is reduced to a one-bit comparator, which simply outputs the sign of the error. Initially, the error is assumed to be equal to one-half the size of the MSB. Once the sign of the error is changed, the amplitude of the error is also reduced by

half. This allows for a binary tuning of the frequency, which is very fast. This is continued until the LSB is reached. At this point, the error is always assumed to be only one LSB. This is valid, since, once the PLL is locked, the error produced by the PFD can be assumed to be very small. A state machine illustrating this method is shown in Figure 6-3. The "shift right" statements refer to reducing the size of the error by one-half.

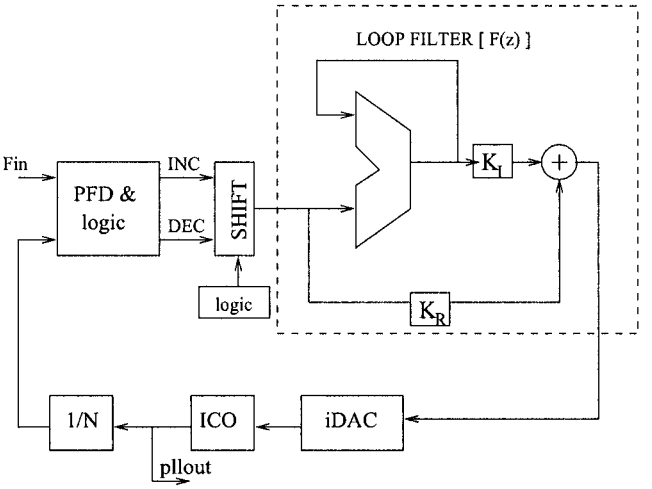


Figure 6-2. DPLL Architecture

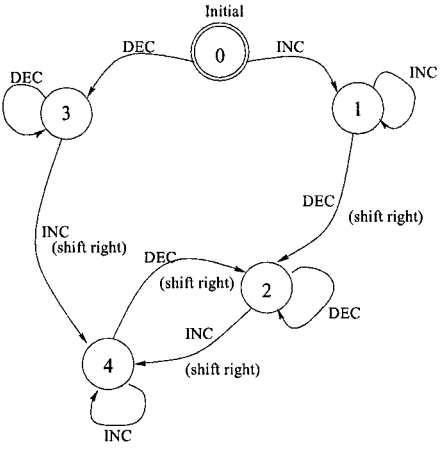


Figure 6-3. State Machine of Locking

The loop filter is updated only when the PFD outputs are updated. When the system is locked, this should be the same rate as the reference frequency. Once the loop filter is updated, the new value is sent to the current digital-to-analog converter (iDAC). The iDAC injects current into the current

controlled oscillator (ICO), as shown in Figure 6-2. The iDAC consists of an array of current sources that are digitally controlled to be on or off.

Note that loop filter parameters K_R and K_I can be used to control the DPLL closed loop bandwidth. For PLLs using ring oscillators, much of the jitter would be due to ICO noise. Generally speaking, the wide loop bandwidth is desired to minimize ICO noise contribution to jitter. However, if the loop bandwidth is too wide, jitter due to the input signal may start to dominate.

In this implementation, an adaptive loop bandwidth mechanism is used to minimize jitter. During locked condition, the PFD should not produce several consecutive INC or DEC pulses. Jitter is estimated by counting the number of consecutive INC and DEC pulses. The values of K_R and K_I are adjusted to optimize the damping factor and bandwidth, respectively.

6.3 Z-domain Analysis

6.3.1 Linear Model of DPLL

When locked, the DPLL behaves as a linear discrete-time system. Figure 6-4 shows a linear model of the DPLL. The iDAC is modelled as a zero-order hold filter (ZOH) [27]. F(z) is the loop filter transfer function, which is given as

$$F(z) = K_R + \frac{K_I}{1 - z^{-1}} \tag{6.1}$$

where K_R and K_I are the forward path and accumulator path coefficients, respectively. The closed loop expression of the DPLL transfer function is given as

$$H(z) = \frac{K \cdot (K_R + K_I) \cdot \left[Z - \frac{K_R}{K_R + K_I} \right]}{Z^2 - \left(2 - \frac{K \cdot (K_R + K_I)}{N} \right) \cdot Z + \left(1 - \frac{K}{N} K_R \right)} \tag{6.2}$$

where K is the forward gain coefficient given as $K_{iDAC} \cdot K_{ico}$, the iDAC and ICO gain coefficients, respectively, and N is the feedback division ratio.

6.3.2 Linear Performance Evaluation

First, the simple case of $K_I = K_R = 1$ is considered. The root locus of the system given by equation (1) is shown in Figure 6-5. As shown, the system is unconditionally stable with a zero at 0.5 on the z-plane. As the open loop

dc gain increases, the poles migrate towards the origin. At this point, the forward gain is equal to 2.

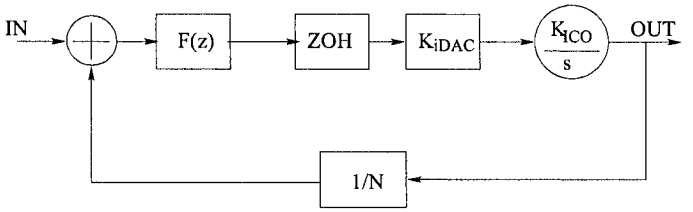


Figure 6-4. Linear discrete-time model of PLL

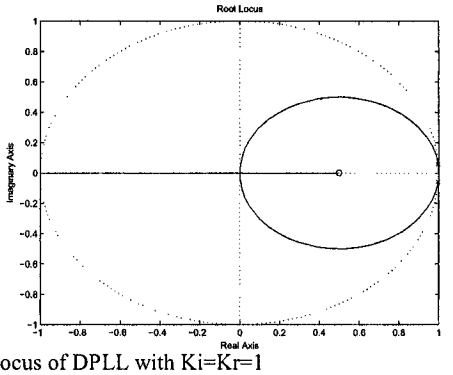


Figure 6-5. Root Locus of DPLL with Ki=Kr=1

As the value of K_R is increased, the size of the circular radius of the pole movement shrinks, as shown in Figure 6-6. It is evident from the figure, the K_R can be effectively used to control the damping factor, and hence, the locking characteristics of the DPLL. Therefore, K_R can be used to control the damping factor of the DPLL. K_R has been found to have small impact on the closed loop bandwidth of the DPLL.

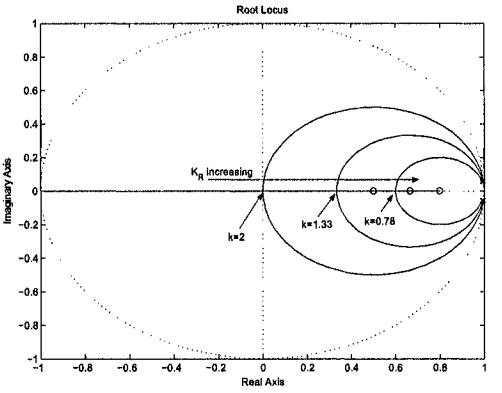


Figure 6-6. Effect of Kr on loop dynamics

K_I affects both the loop bandwidth and damping factor. As K_I is increased, the circular radius of the pole movement is also increased, as shown in Figure 6-7. As K_I is increased the system will be more oscillatory and the positions of the pole would move faster along the z-plane. This means that the closed loop bandwidth of the DPLL is a strong function of K_I.

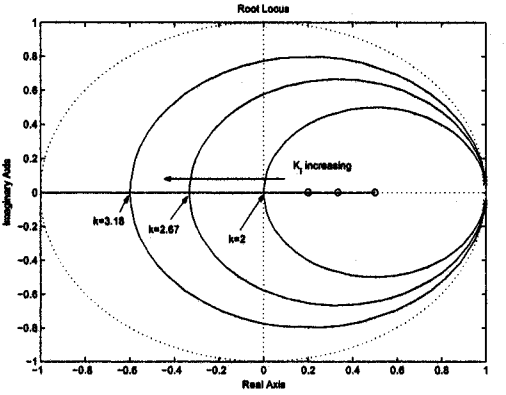


Figure 6-7. Effect of K_I on loop dynamics

The delay through the digital filter is also important. Any delay produces a phase shift, which degrades the performance of the DPLL. Figure 6-8 shows the steady state oscillation amplitude with delay through the digital circuitry. As technology scales, the delay through the digital circuitry would reduce, enhancing the performance of the DPLL.

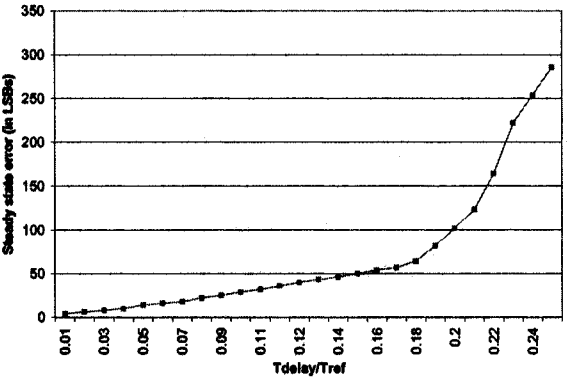


Figure 6-8. Variation of steady state oscillation with loop delay

6.4 Circuit Implementation Issues

In this section, key circuit blocks are given. The digital logic and high-speed feedback divider were implemented using a standard cell library. High-speed addition logic, such as carry-look ahead adders (CLA) [3] was used to implement the loop filters.

6.4.1 DAC optimization

The 10-bit iDAC was partitioned into two segments, as shown in Figure 6-9. This was done to reduce power consumption. When the PLL is locked, only a few bits of the LSB will toggle to track small jitter variations over time, while the MSB bits will be fixed. For this reason, the MSB bits have been made single-ended, while the LSBs were made current steering. Single-ended current sources have the advantage of dissipating zero current while off. Current steering current sources have the advantage of having low switching ripple (and hence less jitter). Cascode PMOS current sources were used in both the MSB and LSB portions of the iDAC to enhance DAC linearity and to provide good immunity to low frequency supply noise. The MSB and LSB DAC arrays were thermally decoded and interdigitated to reduce gradient linearity errors [4]. Dummy devices around the bias transistors were also used for better matching [10]. An example of interdigitation of bias transistors of the 3 LSB binary coded current sources with dummy devices is shown in Figure 6-10.

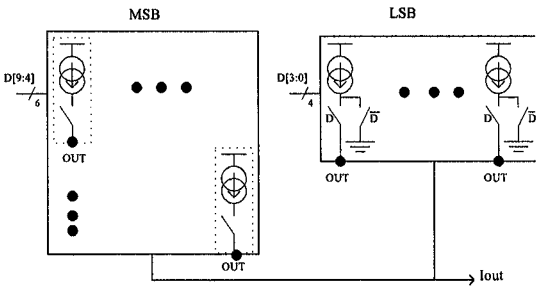


Figure 6-9. iDAC partitioning for low-power

When the DPLL is locked, a few LSB bits can toggle to compensate for any jitter induced by the ICO or externally injected noise. If this toggling is restricted to the LSB section of the iDAC, this should contribute minimally to a reference spur. However, if one MSB current source toggles, this can

cause severe periodic rippling on the ICO input current, which in turn, would produce large reference spurs.

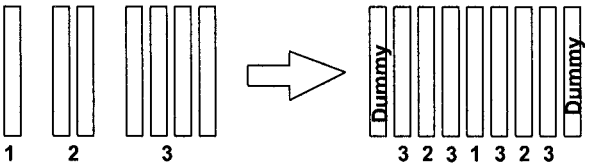


Figure 6-10. Interdigitation of transistors in current mirrors for good matching performance

One way to prevent MSB bits from toggling is to have sufficient bits allocated to the LSBs to cover any variation in the control word due jitter. However, if the steady-state digital filter output with no jitter is at a MSB boundary (i.e. LSBs=00...000 or 11...1111) or even near a MSB boundary, then additional jitter can toggle a single-ended MSB current source.

The correct prevention of MSB toggling can be implemented by using a detect-and-shift approach. Additional logic circuit is added to detect near MSB boundary condition. If the digital filter output is near an upper MSB boundary (i.e. LSBs=11...1111), then ½ MSB is subtracted from the digital filter output and a dedicated current steering current source with weight of ½ MSB adds current into the ICO. If the digital filter output is near a lower MSB boundary (i.e. LSBs=00...0000), then ½ MSB is added to the digital filter output and a dedicated current steering current source with weight of ½ MSB subtracts current from the ICO.

A three-stage ring oscillator was used for the ICO. The delay stages consisted of differential pairs with PMOS symmetric loads [5]. This configuration was chosen for its good immunity to supply noise and its high linearity range, as demonstrated in Chapter 4. The PMOS current sources from the iDAC are dumped directly into the ICO. No additional filtering is required between the iDAC and the ICO. The ICO delay cell used is shown in Figure 6-11.

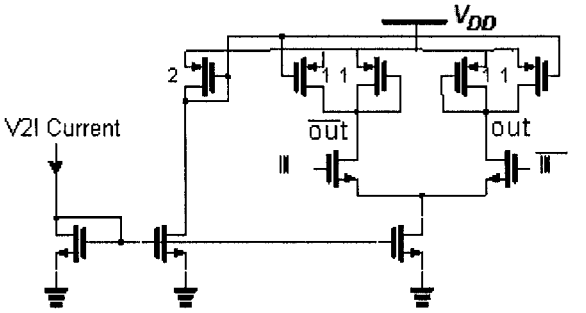


Figure 6-11. ICO delay cell used in DPLL

6.4.2 Reference current temperature stability

One important aspect of the implementation of the DPLL is how to bias the ICO and the iDAC. The current DAC is simply a current mirror with a current gain ratio controlled by the digital input control word. The ICO's frequency of oscillation and gain both depend on the applied input current. The ICO's oscillation frequency is governed by the large-signal gain G_m of the input differential pair transistors in the ICO delay cell. As the temperature of the chip increases, the mobility of the transistors is affected. The mobility of FET devices is proportional to $T^{-3/2}$ [8]. Since G_m is proportional to mobility (see equation 3.6) and device f_T is proportional to G_m (see equation 3.14), it is expected that the speed of the ICO will degrade with increasing temperature.

One way to decrease the sensitivity of ICO performance to temperature is to use a PTAT current reference for the iDAC. Since a PTAT has a positive temperature coefficient, the current produced will increase with temperature and hence some performance loss is compensated for. The slope of the temperature sensitivity of the PTAT can be increased by subtracting off a bandgap current from the PTAT current as shown in Figure 6-12 [9].

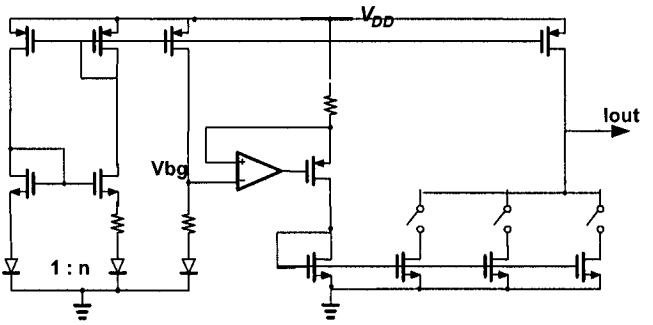


Figure 6-12. "More than PTAT" current reference.

Another method to track the performance of the ICO is to use a constant-gm current source [10]. Figure 6-13 shows a constant-gm circuit. The currents in the two branches are set equal to each other by the diode connected MP1 and MN2 transistors. Since the gates of transistors MN1 and MN2 are shorted, it follows that

$$V_{GS,MN2} = V_{GS,MN1} + I \cdot R \tag{6.3}$$

This equation assumes that the currents in the two branches are identical and there is no transistor mismatches in the two branches. Using equation (3.11), equation (6.3) can be rewritten as

$$\frac{2 \cdot I}{g_{m,MN2}} + V_{T,MN2} = \frac{2 \cdot I}{g_{m,MN1}} + V_{T,MN1} + I \cdot R \tag{6.4}$$

Neglecting the body effect, and assuming that the V_T of transistors MN1 and MN2 are identical, equation (6.4) can be simplified to

$$2\left(\frac{1}{g_{m,MN2}} - \frac{1}{g_{m,MN1}}\right) = R \tag{6.5}$$

It follows from equation (3.12) that keeping the current constant, the transistor transconductance, g_m, is proportional to the square root of the device width. Assuming that the lengths of transistors MN1 and MN2 are equal, it follows that when

$$W_{MN1} = 4 \cdot W_{MN2} \tag{6.6}$$

equation (6.5) reduces to

$$\frac{1}{g_{m,MN2}} = R \tag{6.7}$$

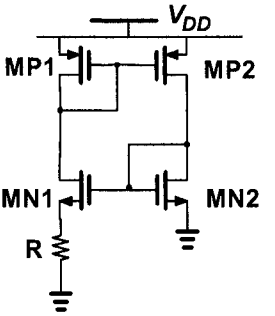


Figure 6-13. Constant gm current reference.

If resistor R is well controlled, then the constant-gm current source produces a current that maintains a constant g_m for transistor MN1. A well controlled resistor R can be obtained by using an external precision resistor or by using a digitally tuned resistor [11].

The constant-gm current source achieves temperature stability as well as stability across process and supply voltage variation. In order to properly use a constant-gm source in the context of an ICO, the dimensions of transistor MN2 must have similar, if not the same, dimensions as the input pair transistors of an ICO delay stage. Transistors MP1 and MP2 are sized

for maximum output impedance as to minimize current mismatches due to differences in V_{DS}. Cascode devices may also be used to help increase minimize current mismatch in the two branches.

6.4.3 Experimental Results

The DPLL has been fabricated in a 0.25um CMOS technology. The frequency spectrum of the DPLL is shown in Figure 6-14. As shown, the close-in phase noise is less than –87dBc/Hz and the closed-loop bandwidth is around 2MHz with a reference frequency of 19.2MHz. Low phase noise has been achieved due to several factors. First, the phase-detector is a simple one-bit phase-detector, as opposed to a large counter or delay line. This greatly reduces the phase-detector's contribution to close-in phase noise. Secondly, a pure digital loop filter was used. This means that the device noise of loop filter's transistors do not contribute to any jitter. Thirdly, a current DAC with cascaded current mirrors were used. The close-in current noise contribution of the iDAC is small since large sized FET devices were used (and hence lower flicker noise). Lastly, the voltage-to-current (V2I) converter has been completely eliminated. This means that noise produced by the V2I is completely eliminated.

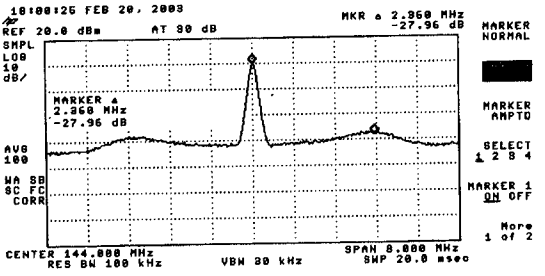


Figure 6-14: Frequency spectrum of DPLL

Figure 6-15 shows the jitter histogram plot of the DPLL. As shown, the DPLL exhibits only 130psec cycle jitter under noisy conditions (microprocessor on). The significant fact to note here is that the histogram of the jitter follows a Gaussian distribution. This means that it contains very little spurious content even when the microprocessor is enabled. This shows another added benefit of the DPLL, which is added noise immunity. As mentioned in chapter 4, if a PLL with dedicated power supplies is used, substrate noise can directly modulate the ICO through the loop filter capacitors. In a DPLL, the loop filter is entirely digital. Moreover, the current sources of the iDAC as well as the current reference generator are all

referenced to the supply voltage. This provides added immunity to substrate noise, enhancing the DPLL's performance in a noisy environment, such as a SoC processor.

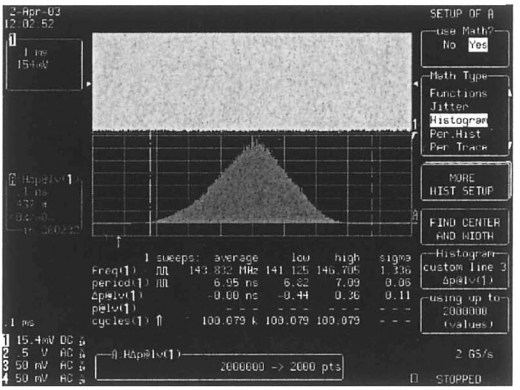


Figure 6-15: Jitter plot of the DPLL

The DPLL performance is summarized in Table 1. As the table shows, the power consumption of the DPLL is very small compared to a typical analog PLL implementation. This has been attributed to the fact that much of the overhead circuitry associated with an analog PLL, such as various bias references, can be completely eliminated. The charge pump, which can consume tens or hundreds of microamperes is substituted with a segmented iDAC, where the MSBs (large currents) are single ended and are turned off when not used. The voltage-to-current converter (and its associated current consumption) has also been completely eliminated.

Table 6-1: DPLL performance summary

Technology	0.25um CMOS
Power Supply	2.6V
Power Consumption	3.12mW @ 144Mhz
VCO Range	40MHz – 160MHz
Cycle Jitter	130ps
Rms Jitter	60ps

Table 2 compares the normalized results of the proposed DPLL with 4 other PLLs. The first two columns are recent DPLL implementations. The third column is an analog-based PLL that has been implemented in the same technology. The fourth column is an analog-based PLL found in the literature that has excellent performance. For a fair comparison, the power has been normalized to frequency (mW/MHz), the area has been normalized to the technology (assuming square reduction of area with technology), and

jitter has been normalized to the square root of the power consumption. The figure-of-merit (FOM) used to compare all the PLL architecture is given as:

$$FOM = \left[\frac{area(mm^2)}{(tech/0.25)^2}\right]\left[\frac{mW}{MHz}\right]^{1.5}\left[jitter(ps)\cdot\sqrt{mW}\right]^2 \quad (6.8)$$

Table 6-2: Comparison of proposed DPLL with others

	Proposed	[1]	[2]	APLL	[7]
Area	1	2.79	1.113	3.43	0.64
Pwr	1	9.23	12.11	2.30	5.77
T_L	1	0.61	0.085	3.33	2.40
Jitter	1	1.52	3.324	1.22	1.03
FOM	1	182.3	518.8	17.74	9.46

As Table 6-2 shows, the proposed DPLL outperforms recent DPLL implementations and has comparable performance to analog-based PLL implementations with less power consumption. Less power consumption is in part due to the elimination of overhead circuitry associated with current generation of charge pump currents, voltage-to-current converters, etc.

6.5 Alternate Digital PLL for Clock Generation

As the previous section demonstrated, digital PLLs have great potential for use in SoC processors. They occupy a small area, have reasonable performance, and consume low-power. In this section, the two topologies that are mentioned in Table 6-2 are now explored.

6.5.1 All-Digital PLL for Digitally Controlled Oscillator

PLL digitization can be carried out a step further. The ICO and the iDAC can be merged into one unit. The resulting structure is called a digitally controlled oscillator, DCO. In its simplest form, a ring oscillator-based DCO can consist of three delay stages. One possible DCO delay cell implementation is shown in Figure 6-16. Each delay cell contains current sources which can be switched on or off. The current sources of each delay cell are binary weighted in order to ease the interface between the digital loop filter and the DCO.

The DCO can be used in a generic DPLL shown in Figure 6-17. A phase detector performs the same function as a regular PLL. However, since it interfaces with a digital loop filter, the phase error must be translated from time into digital format. The loop filter is a digital filter of order K and consisting of numerical coefficients. To reduce complexity of the digital filter, the coefficients are limited to powers of two. This helps replace area

consuming digital multipliers with simple shift-and-add circuits. The digital loop filter is usually implemented by a combination of a forward path and an accumulator, as shown in the previous section. This is an example of an infinite impulse response (IIR) filter. In principle, the digital filter can also be a finite impulse response (FIR) filter. In [13], the use of FIR filter in a DPLL loop filter has been demonstrated to limit the overshoot frequency of a DPLL during locking from 100% down to 50%. Other loop filter implementations to control the overshoot have also been attempted [14]. Unconventional digital loop filters have also been designed that contain a notch characteristic to suppress certain interferers at known frequencies [15].

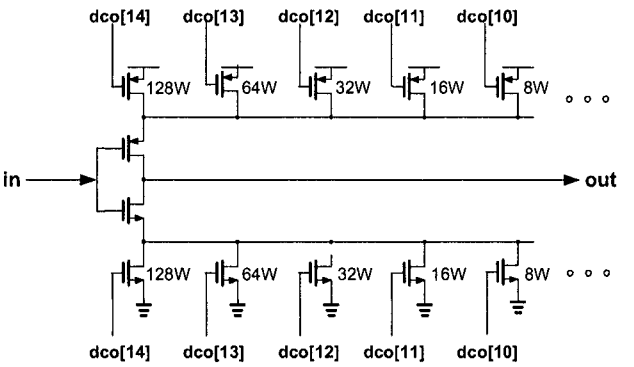


Figure 6-16. Schematic of a DCO

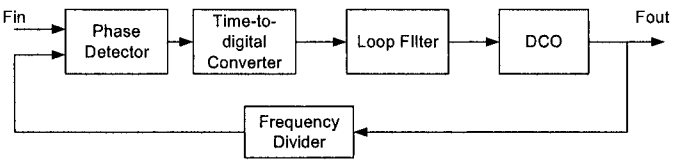


Figure 6-17. Generic DPLL using a DCO

Although the function of a phase detector of a DPLL is the same as a regular PLL, its implementation may be quite different. The phase dector and the time-to-digital converter are usually merged into one block. The simplest implementation of a phase detector is to have two counters in both the feedback, V, and reference, R, path inputs to the phase detector. Upon completion of one of the counters, the content of the other counter contains a digital value proportional to the phase difference between R and V. The sign of the signal is depends on which counter completed first. If the R counter completes first, then the DCO divided-by-V is lagging behind the reference

crystal, and the error is positive. On the other hand, if the V counter finishes first, then the reference signal is lagging behind the DCO divided-by-V signal, and the error is negative.

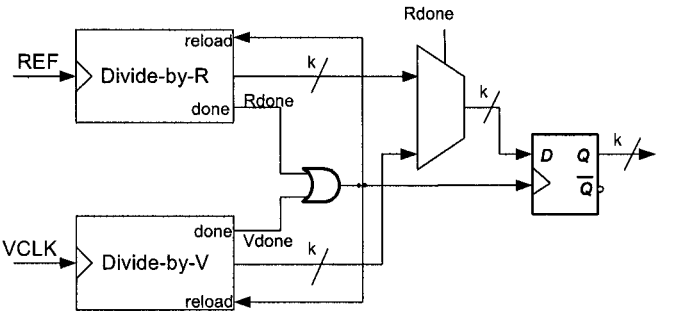


Figure 6-18. Type-1 DPLL phase detector

This type of phase detector is referred to as a type-1 DPLL phase detector. The disadvantage of such a phase detector is that resolution is traded off with loop bandwidth. The phase error produced by the phase detector can be mathematically expressed as

$$\Delta T = T_{DCO} \cdot V - T_{REF} \cdot R \tag{6.9}$$

Upon completion of one of the counters, the contents of the other counter is ideally

$$VAL = \frac{\Delta T}{T_{REF}} \cdot 2^b + b_q \tag{6.10}$$

where b is the number of bits in the counters and b_q is the quantization error in bits. It is assumed here that both the R and V counters are b bits long. Due to quantization errors, the exact value of VAL will differ slightly from that given in equation (6.10). The quantization error can be expressed as

$$\varepsilon_q = \Delta T - \frac{VAL}{2^b} \cdot T_{REF} \tag{6.11}$$

where ε_q is the quantization error in units of time. As indicated by equation (6.11), the quantization error can only be improved by increasing the number of bits b or by increasing the reference frequency T_{REF}. Both reduce the loop bandwdith. For reasonable jitter performance (i.e. less than 250ps) the

reduction is bandwidth forces the lock time to be measured in the milliseconds.

The advantage of the type-1 DPLL phase detector is that a synchronous timing scheme can be used for the DPLL. This means that, at locked condition, the DPLL is clocked at the phase detector at frequency F_{REF}. No additional timing signals are necessary.

A type-2 phase detector is shown in Figure 6-19. A type-2 phase detector does not contain amplitude information, only phase information [2,16]. Generally, a type-2 DPLL phase detector has two outputs indicating and UP, DOWN, or NEUTRAL state. The neutral state can be when both UP and DOWN have the same sign (i.e. both are '1' or '0'). The type-2 DPLL phase detector also has an additional ENABLE , or CKOUT signal. The CKOUT signal is used to clock the digital loop filter. This implies an asynchronous timing scheme, where each block of the DPLL must output a completion signal for the next stage to activate.

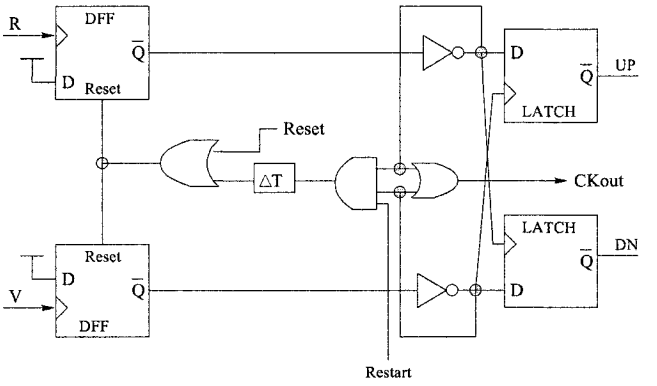


Figure 6-19. Type-2 DPLL phase detector

The type-2 DPLL phase detector implicitly implies that an additional frequency detector is needed for the DPLL to lock in a reasonable time. It also assumes that the error at steady-state is small, since the magnitude of the phase error is always assumed to be unit size. The DPLL described in the previous section uses a type-2 DPLL phase detector with an adaptive phase error amplitude control for better loop dynamics performance. Other amplitude control mechanisms are also possible [17].

6.5.2 All-Digital FLL with Phase Reset

As demonstrated above, the main limitation of DPLLs employing a type-1 DPLL phase detector is the stringent trade-off between phase error accuracy of the phase detector and DPLL lock time. In [1], the type-1 DPLL

phase detector is replaced by a frequency detector for better performance with small lock time.

The all-digital FLL is shown in Figure 6-20. The phase detector is replaced by a digital frequency detector. The frequency detector is followed by the loop filter consisting of a barrel shifter and an accumulator. At every sign change output of the frequency detector the barrel shifter output magnitude is reduce by half, until the output of the barrel shifter, K_I, is 1. Since a FLL is employed only an accumulator is needed. This is analogous to having a PLL with only a capacitor in the loop filter. In this case, however, the pole introduced by the ICO is cancelled by the frequency detector, and loop stability can still be maintained with only an integrator in the loop filter.

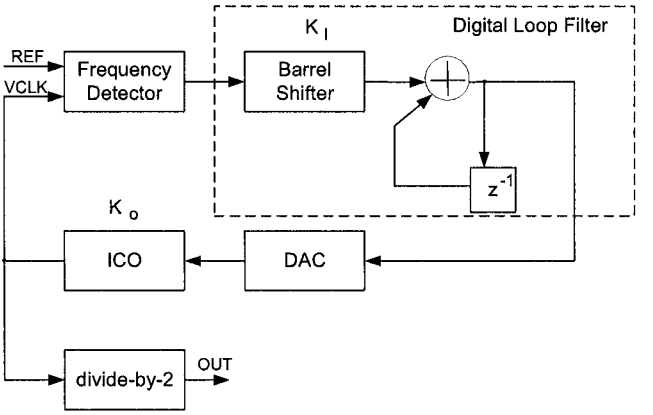


Figure 6-20. All-digital FLL block diagram

As in a type-2 DPLL phase detector, the output of the frequency detector output contains to amplitude information. It is assumed that at locked condition, the phase detector output is small enough such that the frequency detector can still be modeled as a linear block. However, unlike a PLL employing a type-2 DPLL phase detector, the loop filter can be clocked at the same rate as the frequency detector's comparison rate.

With linear assumption of the frequency detector, it is possible to derive a closed loop expression of the FLL discrete-time frequency response. A closed loop expression for the FLL can be given as

$$H(z)=\frac{K\cdot z^{-1}}{1-(1-K)\cdot z^{-1}} \tag{6.12}$$

where

$$K = \frac{K_I \cdot K_o}{\omega_{REF}} \tag{6.13}$$

and K_I is the frequency detector gain, K_o is the ICO gain, and ω_{REF} is the input reference angular frequency (rads/sec). It is important to note that for stability, the condition of $0<K<2$ must be satisfied.

A diagram of the frequency detector is shown in Figure 6-21. It consists of two main blocks: a sampler and a 6-bit synchronous counter. The sampler is used to enable the synchronous counter according to the timing diagram shown in Figure 6-22. Two phases of the input REF signal are generated. The synchronous counter is reset at every ϕ_2 period. Resetting a counter is mathematically equivalent to differentiating the output. This means that instead of producing a $\Delta\theta$ output, a $\frac{\Delta\theta}{\Delta T} = \Delta f$ term is produced. Hence the output is the frequency difference, and not the phase difference, between REF and VCLK.

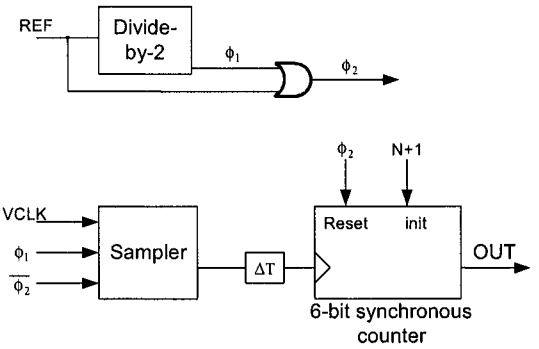


Figure 6-21. Digital frequency detector block diagram

One advantage of an FLL is its ability to operate as a synchronous system. The timing diagram for synchronous FLL operation is shown in Figure 6-22. One FLL cycle operates over two REF cycles. During the first REF cycle, the FLL measures the frequency difference between the REF and VCLK outputs. During the second cycle the loop filter is updated and the ICO is reset with the REF signal in order to cancel out any phase errors between the REF and VCLK signals. This last step is what enables phase lock. The disadvantage of this approach is that the ICO is constantly perturbed and must recover quickly in order to minimize the jitter caused by perturbation. The reset operation of the ICO has the advantage of limiting jitter accumulation in the ICO; however, as the overall results in [1] indicate, the overall jitter of the FLL is not much different from other state-of-the-art fast-lock digital PLLs.

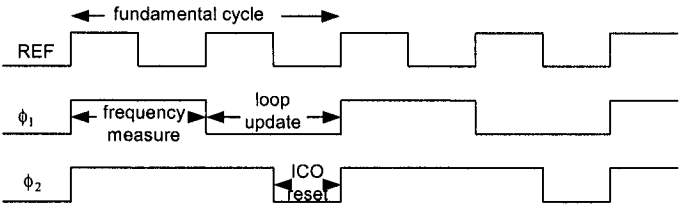


Figure 6-22. Timing diagram for synchronous FLL operation

6.6 Summary

In this chapter, an alternative paradigm in PLL design is given. The loop filter is digitized. This presents certain advantages and design challenges. A low-power, area efficient digital PLL (DPLL) architecture was presented. The entire loop filter was replaced by a digital filter consisting of adders, shifters, and registers. A current-mode digital-to-analog (iDAC) was used to interface the digital loop filter with an ICO. An asynchronous clocking mechanism was used to prevent excessive delay within the loop filter, which may otherwise result in instability. The DPLL achieves fast lock by using a binary locking algorithm.

The implementation of this DPLL was also detailed, with emphasis on the iDAC, which is an enabling technology for the DPLL. Matching between current sources in the iDAC is important. Matching was achieved by using a segmented DAC architecture as well as careful layout techniques such as interdigitation and the use of dummy devices.

The overall experimental results prove the viability of the DPLL. Low-jitter was enabled by the fact that much of the sensitive analog circuitry has been digitized and an adaptive loop bandwidth algorithm has been implemented. Low-power has also been achieved by circuit optimizations in the iDAC and by eliminating much of the power consuming analog circuitry.

Alternative DPLL approaches have also been reviewed. The use of a DCO entails the merging of the iDAC and ICO into one structure. An alternative architecture in which a frequency locked loop (FLL) is used instead of a PLL. In order to compensate for any phase errors, the FLL resets the phases of both the phase reference signal and the feedback signal by using a counter and performing an asynchronous reset at frequency lock.

REFERENCES

[1] I. Hwang, et. al., "A Digitally Controlled Phase-Locked Loop with a Digital Phase-Frequency Detector," *IEEE J. of Solid-State Circuits*, vol. 36, no. 10, pp. 1574-1581, October 2001.

[2] C. Chung and C. Lee, "An All-Digital Phase-Locked Loop for High-Speed Clock Generation," *IEEE J. of Solid-State Circuits*, vol. 38, no. 2, pp. 347-351, February 2003.

[3] A. Bellaouar and M. Elmasry, *Low-Power Digital VLSI Design*, Boston: Kluwer Academic Publishers, 1995.

[4] Tien-Yu, W. et. al., "A Low Glitch 10-bit 75MHz CMOS Video D/A Converter," *IEEE JSSC*, vol. Jan. 1995, pp. 68-72.

[5] Maneatis, J., "Low-jitter and process independent DLL and PLL based on self biased techniques," *ISSCC*, 1996, pp. 130-131.

[6] A. M .Fahim, "A Low-Area, Low-Power Digital PLL Clock Generator," *European Solid-State Circuits Conference*, pp. 101-104, September 16-18, 2003.

[7] H. Ahn and D. Allstot, "A Low-Jitter 1.9-V CMOS PLL for UltraSPARC Microprocessor Applications," *IEEE J. of Solid-State Circuits*, vol. 35, no. 3, pp. 450-454, March 2000.

[8] Y. Taur and T. Ning, *Fundamentals of Modern VLSI Devices*, Cambridge: Cambridge Press University, 1998.

[9] G. Rincon-Mora, *Voltage References*, New Jersey: IEEE Press, 2002.

[10] B. Razavi, *Design of Analog CMOS Integrated Circuits*, Boston: McGraw Hill, 2001.

[11] A.Hastings, *Art of Analog Layout*, New Jersey: Prentice-Hall 2000.

[12] R. Holzer, "A 1V CMOS PLL Designed in High-Leakage CMOS Process Operating at 10-700MHz," *Int'l Solid- State Circuits Conference*, pp. 16.5, 2002.

[13] F. Kobayashi and M. Haratsu, "A Digital PLL with Finite Impulse Responses," *Int'l Symposium on Circuits and System*, vol. 1, pp. 191-194, 1995.

[14] Z. Lin, J. Wu, X. He, "Digital PLL with Controllable Frequency Response Time and Overshoot," *INTELEC*, pp. 321-325, 2001.

[15] T. Inoue, et. al., "Interference Suppression Using DPLL with Notch Frequency Characteristic," *Int'l Symposium on Circuits and Systems*, pp. 2084-2087, 1989.

[16] J. Chiang and K. Chen, "A 3.3V All Digital Phase-Locked Loopo with Small DCO Hardware and Fast Phase Lock," *Int'l Symposium on Circuits and Systems*, vol. 4, pp.554-557, 1998.

[17] N. Kim, "Design of ADPLL for Both Large Lock-IN Range and Good Tracking Performance," *IEEE Trans. On Circuits and Systems II*, vol. 46, no. 9, pp. 1192-1204, September 1999.

[18] T. Olsoon, "An all-Digital PLL Clock Multiplier," *IEEE Custom Integrated Circuits Conference,* pp. 275-278, 2002.

[19] R. Fried, "Low-Power Digital PLL with One Cycle Frequency Lock-In Time for Clock Syntheses up to 100MHz Using 32,768 Hz Reference Clock," *Ninth Annual IEEE ASIC Conference and Exhibit*, pp.291-294, 1996.

[20] R. Fried, "High-efficiency low-power on-clock solutions for multi-clock chips and systems," *IEEE-CAS Workshop on Analog and Mixed IC Design*, pp. 60-65, 1996.

[21] J. Lorenzo-Ginori and J. Naranjo-Bouzas, "All-Digital PLL with extended tracking capabilities," *Electronics Letters*, vol. 33, no. 18, 28 August 1997, pp. 1519-1521.

[22] T. Watanabe and S. Yamauchi, "An All-Digital PLL for Frequency Multiplications by 4 to 1022 with Seven-Cycle Lock Time," *IEEE J. of Solid-State Circuits*, vol. 38, no. 2, pp. 198-204, February 2003.

[23] J. Savoj and B. Razavi, "Design of Half-Rate Clock and Data Recovery Circuits for Optical Communication Systems," *Design Automation Conference*. pp. 121-124, 2001.

[24] M. Ramezani and C. Salama, "An Improved Bang-bang Phase Detector for Clock and Data Recovery Applications," *Int'l Symposium on Circuits and Systems*, vol. 1, pp. 715-718, 2001.

[25] R. Bogdan, et. al., "All-Digital Phase-Domain TX Frequency Synthesizer for Bluetooth Radios in 0.13um CMOS," *Int'l Solid-State Circuits Conference*, pp. 272-273, 2004.

[26] J. Dunning, et. al., "An All-Digital Phase-Locked Loop with 50-Cycle Lock Time Suitable for High-Performance Microprocessors," *IEEE J. of Solid-State Circuits*, vol. 30, no. 4, pp. 412-422, 1995.

[27] J. Hein and J. Scott, "z-Domain Model for Discrete-Time PLL's," *IEEE Trans. On Circuits and Systems*, vol. 35, no. 11, pp. 1393-1399, November 1988.

[28] F. Gardner, "Charge-Pump Phase-Lock Loops," *IEEE Trans. On Communications*, vol. COM-28, no. 11, pp. 1849-1856, November 1980.

[29] T. Olsson and P. Nilsson, "Fully Integrated standard cell digital PLL," *Electronics Letters*, vol. 37, no. 4, pp.211-212, 15th February 2001.

Chapter 7

DSP CLOCK GENERATOR ARCHITECTURES

7.1 Introduction

In the previous chapters, phase-locked loops (PLLs) have been analyzed in both the system and circuit level. Various PLL circuit and architectural topologies have also been analyzed. In this chapter, clock generators employing PLLs are explored. First, the requirements of digital signal processors (DSPs) and data converters in system-on-a-chip (SoC) processors are analyzed in detail. This is followed by a description of variety of clock generating architectures that can be used to obtain the required frequencies for DSPs and data converters. Examples and performance analysis of each architecture is also given.

7.1.1 Why important?

Advances in semiconductors have allowed engineers to shrink transistor sizes enabling more on-chip digital circuitry to be integrated onto a single chip. The cost sensitive market of the wireless telecommunication industry, for example, has also forced designers to cut costs of microchips. For this reason, dedicated microchip solutions are often used in telecommunication market that integrate a general-purpose processor, such as an ARM processor, with application specific hardmacros, such as a Fast Fourier Transform (FFT) unit, as well as analog circuitry, such as PLLs and data converters. More recently, the integration of key analog blocks, such as data converters, has become pivotal to reducing cost and power consumption of microchips.

It may be necessary, however, to integrate a variety of signal processing functions onto a single SoC device. Any signal processing device (analog or digital) requires a very specific sampling rate. For example, consider a system that requires CD stereo quality audio input/output with a sampling rate of 44.1kHz, DVD audio and video sampling rates of 96kHz, an 802.11a wireless LAN sampling rate of 20MHz, and a general-purpose microprocessor frequency of 100MHz. Having a single PLL producing all required frequencies can be challenging. Each application has its own requirements in terms of time-domain total jitter as well as frequency-domain spectral purity requirements.

7.1.2 Typical DSP Clock Generation Specifications

In SoC processors, an on-chip PLL is used as a sampling clock for the various signal processing blocks. Important specifications of signal processing blocks include multiply-and-accumulates (MACs) per second, spurious free dynamic range (SFDR), and signal-to-noise ratio (SNR). The first specification is related to digital signal processing (DSP). Computing the output of any cascaded blocks of linear systems involves the multiplication of the transfer function of block in the frequency domain, or convolution in the time domain. Assuming that the system is a finite-impulse response (FIR) system [1], the discrete-time version of convolution can be expressed as

$$c_n = \sum_i a_i \cdot b_{N-i} \tag{7.1}$$

where a_i and b_i are the coefficients of the two discrete-time system A(kT) and B(kT). Note that a_i and b_i can also represent a discrete-time system A(kT) and input samples b_i. As shown above, the calculation of output samples c_n involves a multiplication followed by a sum, repeated several times. The MACs per second rating of a DSP thus upper bounds the maximum rate of the digital input samples, or sampling rate of the system.

7.2 Sampling Clock Requirements for Data Converters

7.2.1 First order Analysis

The spurious free dynamic range (SFDR) and signal-to-noise ratio (SNR) metrics, mentioned above, are typically used to quantify the performance of analog-to-digital (A/D) and digital-to-analog (D/A) converters. In A/D

converters, jitter has the effect of falsely triggering the sample time. Consider an input sine wave shown in Figure 7-1. Uncertainty in sampling time due to clock jitter causes an amplitude measurement error of ΔA.

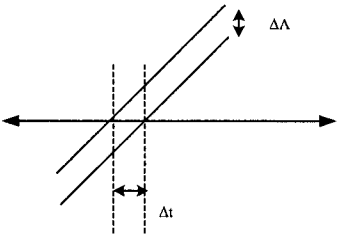


Figure 7-1. Effect of jitter on A/D input signal

If the input signal is sinusoidal, it can be mathematically described as

$$C(t) = A \cdot \cos(2\pi ft) \tag{7.2}$$

where A is the peak amplitude and f is the frequency of the sine wave. Differentiating equation (7.2) will give the slope of the curve. The amplitude is maximum at zero crossing, and in terms of equation (7.2), this occurs when cos(2πft)=1. This means that the maximum slope is

$$\frac{\Delta A}{\Delta t} = A \cdot 2 \cdot \pi \cdot f \tag{7.3}$$

where Δt is the jitter. The SNR of the A/D converter is simply given as ΔA/A and it follows that for a given SNR requirement, the maximum clock jitter can have is given as

$$\sigma_{rms} = \frac{1}{2\pi \cdot f \cdot SNR} \tag{7.4}$$

The above analysis assumes that the jitter in the clock generator follows a Gaussian distribution. In general, the clock generator can contain spurious components. Spurious content can result from periodic noise injected into the clock signal through the supply voltage or nearby digital circuitry. Spurious content may also result if a fractional-N PLL is used, as discussed in section 7.3. Since the clock generator samples the data converter input signal, the output of this sampling process is actually a time-domain multiplication of the input signal with the clock signal. If the input clock signal contains frequency spurious content, one of two conditions may occur, as shown in Figure 7-2(a) and (b). In the first case, the frequency spur is well beyond the input signal bandwidth. The frequency spur does not degrade the data converter's SNR. In the second case, however, the

frequency spur is within the input signal's bandwidth. The maximum SNR in this case is equal to the ratio between the amplitude of the signal to the spur's maximum amplitude. In other words, the SNR is given as

$$\mathrm{SNR} = \frac{A}{\sum_i A_{\mathrm{spur},i}} \tag{7.5}$$

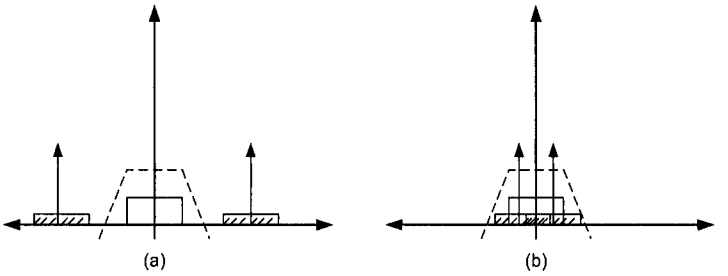


Figure 7-2. Effect of clock spurs on output spectrum for (a) out-of-band spurs and (b) in-band spurs

The actual SNR may actually be better than what (7.5) predicts depending on the activity of the input data. For example, consider the case when the reference spur is at 20kHz and the data is an analog voice signal. Analog voice signal is sampled at 8kHz. In this case, the signal modulated by the spur does not interfere with the original baseband signal and the effect of the spur is minimal. Now, consider the case where the input signal also contains music. Music has a bandwidth of 20kHz. In this case, the spur may significantly degrade the data converter's SNR.

7.2.2 General Case – A Qualitative Approach

In the above analysis, the simple case of a single sinusoidal input has been assumed. In general, any kind of input waveform can be applied to the data converter. The power spectral density of the PLL clock jitter is not white, but shaped by the VCO jitter, clock reference jitter, and the PLL transfer function. As more functions are integrated on-chip, more noise can be coupled onto the clock lines and more stringent requirements can be imposed on the data converters. These two contradicting trends require a more rigorous analysis of the effects of clock jitter on data converters, and more specifically analog-to-digital converters (ADCs).

In essence, an analog-to-digital converter (ADC) is a sampled system. The sampling action itself can be modeled as a mixing operation as long as the phase jitter is small compared to the inverse of the maximum input signal

frequency content. In this case, small angle approximation of $\sin(\theta) \approx \theta$ can be applied. Mixing, or multiplication, in the time-domain is convolution in the frequency domain. Therefore, the power spectral density of the output can be expressed as

$$S_y(f) = S_x(f) * [1 + S_\phi(f)] \tag{7.6}$$

where $S_y(f)$ is the output power spectral density, $S_x(f)$ is the input data power spectral density, and $S_\phi(f)$ is the phase noise spectrum of the clock signal. Note that if the input clock is ideal, the output spectrum is equal to the input spectrum. This mixing operation in the frequency domain is shown in Figure 7-3. An input clock with Gaussian as well as deterministic jitter has been assumed.

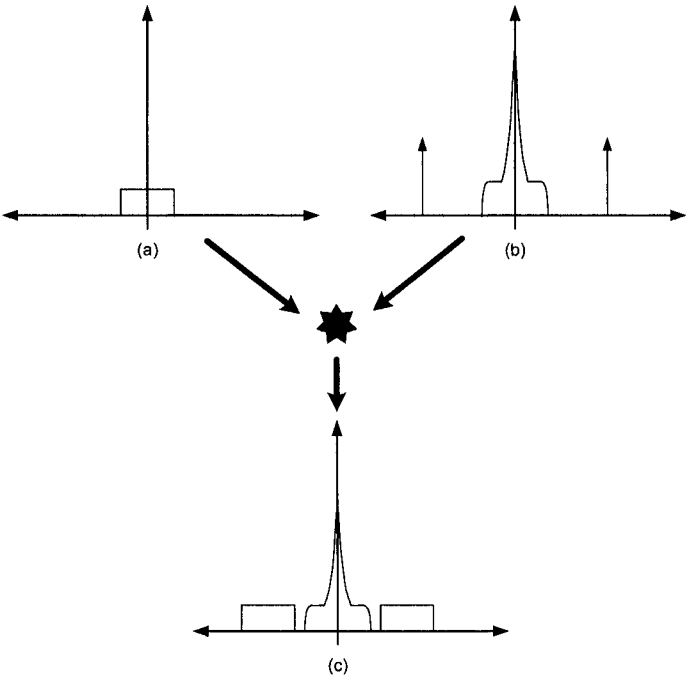


Figure 7-3. Spectrum of A/D converter with noisy input clock: (a) input signal PSD, (b) clock phase noise, (c) output signal PSD.

7.2.3 General Case – A Quantitative Approach

The general model given above can also be expressed analytically [4]. Linearizing the input signal at the sampling time, gives an expression for an output signal as

$$y(kT) = x(kT) + x'(kT) \cdot t_j(kT) \tag{7.7}$$

where y(kT) is the discrete-time output, x(kT) is the discrete-time output, $x'(kT)$ is the first derivative of the discrete-time input signal, and $t_j(kT)$ is the accumulated jitter at sampling time. Since inverse Fourier transform of the power spectral density is the autocorrelation function, it is instructive to calculate the autocorrelation function of the output. The autocorrelation function of the output can be given as

$$r_y(mT) = E[y(kT + mT) \cdot y(kT)] \tag{7.8}$$

Substituting (7.7) into (7.8) gives

$$r_y(mT) = r_x(mT) + r_{x'}(mT) \cdot r_{tj}(mT) \tag{7.9}$$

where r_x and r_{tj} are the autocorrelation functions of the input and clock jitter, respectively. With further manipulation, it can be shown that the autocorrelation function of the first derivate of the input is equal to the second derivative of the autocorrelation function of the input. In other words,

$$r_{x'}(mT) = -r_x''(mT) \tag{7.10}$$

where r_x'' is the second derivative of the autocorrelation function. Substituting (7.10) into (7.9) and taking the Fourier transform gives

$$S_y(f) = S_x(f) + \left[(2\pi f)^2 S_x(f)\right] * S_{tj}(f) \tag{7.11}$$

where S_{tj} is the power spectral density of the clock jitter. Using phase noise instead of S_{tj} in (7.11), the $2\pi f$ term drops out and the resulting expression is the same as equation (7.6). This means that this analytical derivation also assumes small angle approximation of phase jitter.

The A/D data converter signal-to-noise (SNR) ratio can be expressed as the ratio of the total energy of the input signal to the total energy of the error signal. Substituting (7.10) into (7.9) and subtracting the input signal from (7.9) gives the autocorrelation function of the error signal, which is

$$r_e(mT) = -r_x''(mT) \cdot r_{tj}(mT) \tag{7.12}$$

Since the energy of a signal can be expressed as its autocorrelation function at time 0, the maximum signal-to-noise ratio due to clock jitter can be expressed as

$$SNR = \frac{E_{signal}}{E_{error}} = \frac{r_x(0)}{-r_x''(0) \cdot r_{tj}(0)} \tag{7.13}$$

For the simple case of an input sine wave, it can be easily shown that equation (7.13) leads precisely to the expression given in equation (7.4).

7.2.4 Specific Case – Sigma-Delta A/D Converters

The above analysis is true for Nyquist rate A/D converters (ADCs). Nyquist rate data converters sample the input signal at twice the maximum input data rate. For oversampled sigma-delta data converters, however, the output spectrum is more sensitive to input spurs.

Sigma-delta data conversion is a technique that is widely used to achieve high resolution using low quality electronics [19-22]. Typically, a one-bit quantizer is used as shown in Figure 7-4. The operation of sigma-delta converters is discussed in more detail in section 7.4.

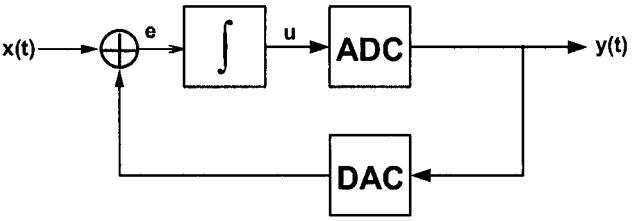


Figure 7-4. Sigma-delta ADC

As the figure shows, the input is first applied to a difference node that subtracts the input signal from the output of the DAC in the feedback path. The resulting signal, e, is the error between the input signal and the predicted signal at the output. The error is then accumulated by the integrator denoted by the $\int$ symbol. It is assumed that the integrator is a continuous-time analog filter. As the summation of the error increases, the magnitude of the output of the modulator increases. This, in turn, decreases the next error signal produced. Therefore, the sigma-delta modulator is a closed loop feedback mechanism that tracks the input to produce an equivalent digital output term. The feedback DAC is only needed to interface between the digital output and the analog input.

The clock signal samples the error terms, e, in the ADC. The resulting discrete-time output spectrum can be expressed as:

$$Y(z) = z^{-1} \cdot X(z) + (1 - z^{-1}) \cdot Q(z) \tag{7.14}$$

where Q(z) is the quantization noise introduced by the ADC. The ADC is

modeled as a summation noise adding the signal from the integrator with the quantization noise. The quantization noise is assumed to be additive white noise with a uniform distribution.

A typical output spectrum from a sigma-delta modulator is shown in Figure 7-5. As shown, the waveform is a high-pass filter with zero content at DC and F_{CLK}. The spectrum repeats at every multiple of F_{CLK} since the signal is sampled in the time-domain. In effect, the sigma-delta shapes the quantization noise to higher frequencies. It is assumed that a low-pass filter following the sigma-delta modulator then suppresses the high-frequency quantization noise. In order to reduce the quantization noise error, the sampling clock is typically run at a much higher frequency than the Nyquist rate. Rates of 40-100 are not uncommon for audio applications.

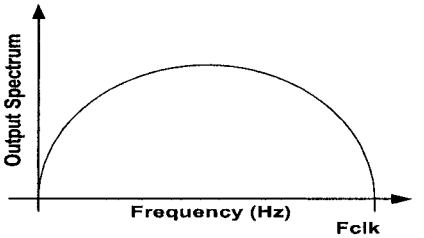


Figure 7-5. Typical output spectrum from a sigma-delta modulator

Assuming that the quantization noise is uniformly distributed white noise, it can be shown that the power spectral density of the output of the modulator, assuming no input x(t), is

$$S_Y(f) = S_Q(f) \cdot \left|\frac{Y(z)}{Q(z)}\right|^2 = (2\sin \pi fT)^2 \frac{\Delta^2}{12} \frac{1}{f_s} \tag{7.15}$$

The most intriguing effect here is the effect of clock jitter on an oversampled sigma-delta data converter. Figure 7-6 shows the power spectrum of the sampled output signal. As before, the clock jitter power spectrum is convoluted with the input signal. In this case, however, the input signal has been shaped by a sigma-delta transfer function, but the high frequency quantization noise has not been low-pass filtered yet.

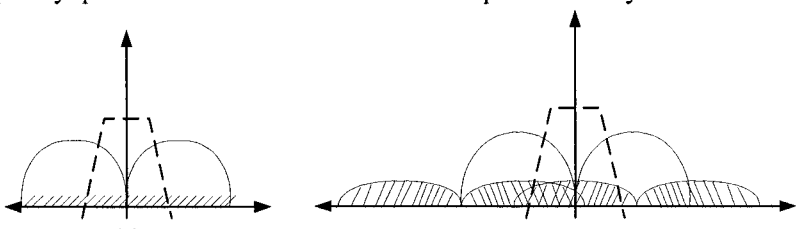


Figure 7-6. Power spectral density with clock jitter being (a) white Gaussian noise and (b) Gaussian noise with spurious signals

In the first case, the input signal is convoluted with Gaussian white noise. The effect is that the noise floor is raised by an amount equal to the clock jitter's power spectral density. Since the input is oversampled, the data bandwidth is a small fraction of the clock frequency, and hence only the noise energy up to the data bandwidth needs to be considered (shown by broken line in Figure 7-6).

In the second case, the clock jitter contains some spurious signals. Each spur will sample the expression shown in equation (7.15) with a center frequency equal to the location of the spur. This can be a serious issue since an out-of-band spur can cause a sample of high-frequency quantization noise to fold back to within the data's bandwidth. This quantization noise will not be filtered by the low-pass filter following the data converter, and hence will lower the data converter's resolution. For this reason, sigma-delta converters are far more sensitive to deterministic clock jitter than to random clock jitter. The resulting output spectrum with deterministic clock jitter can be expressed as:

$$S_Y(f) = \sum_i \left[(2\sin \pi f T)^2 \frac{\Delta^2}{12} \frac{1}{f_s} \right] * [\delta(f - f_i)] \tag{7.16}$$

where $\delta(f)$ is the Dirac function which takes on a value of infinity $\delta(0)$ and zero elsewhere. Summing this expression over the data bandwidth gives the total energy of the quantization noise that will not be filtered by the low-pass filter following the sigma-delta modulator. Dividing the total energy of the signal by this quantity gives the maximum signal-to-noise ratio due to clock jitter. The resulting SNR is

$$SNR = \frac{r_x(0)}{r_e(0)} = \frac{E_{signal}}{\int_0^{BW} \sum_i \left[(2\sin \pi f T)^2 \frac{\Delta^2}{12} \frac{1}{f_s} \right] * [\delta(f - f_i)] \cdot df} \tag{7.17}$$

7.3 Jitter in Frequency Dividers

Although literature on digital frequency dividers exists [9-16], it mostly relies on empirical data to derive models for frequency divider jitter. In this section, a more analytical handling of digital frequency dividers is presented [17].

7.3.1 Uses of Frequency Dividers in SoC Processors

Jitter reduction in clock networks and synthesizers is pivotal to the success of SoC processors. Excessive jitter can violate timing paths, causing

catastrophic failures. The performance of frequency dividers in clock generators and frequency synthesizers is often either ignored or underestimated. In this section, a thorough analysis of jitter due to digital frequency dividers is given under various conditions.

There are two main uses of digital frequency dividers in SoC processors. One use of frequency dividers is as a feedback divider in a PLL. applications, as shown in Figure 7-7(a). Frequency dividers can also be used as a post-divider after a PLL or a crystal oscillator, as shown in Figure 7(b). This type of implementation is most oftenly used in system-on-a-chip (SoC) processors where a variety of frequencies are required for the various units on-chip.

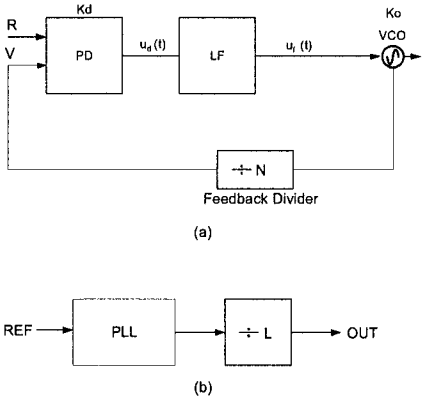


Figure 7-7. Uses of digital frequency dividers: (a) in a PLL and (b) stand-alone

7.3.2 Mechanism of Jitter in Digital Frequency Dividers

A typical output of a digital frequency divider with amplitude noise injected is shown in Figure 7-8. As shown in Figure 7-8(a), amplitude noise causes a time-shift in the zero-crossings of the divider output. This time-shift is a function of the amplitude noise, which is random. Hence the time-shift is also random, or jittery. If noise was injected during the low and high flat regions of the divider's output waveform, the amplitude noise will not translate into phase noise. This indicates that the nature of phase noise, or jitter, in frequency dividers is time-varying (similar to the case of VCO jitter explained in chapter 4).

Since the output waveform of a digital frequency divider (DFD) is also periodic, this indicates the nature of jitter in digital frequency dividers is also cyclostationary. Cyclostationary noise results when the mean and autocorrelation function of the noise source are bounded and periodic [5]. There are three main types of noise arising from silicon devices: shot noise,

thermal noise, and flicker noise [6]. The power spectral density of shot noise and thermal noise are white, and hence their mean and autocorrelation function are bounded. Flicker noise, on the other hand, does not have a bounded mean or autocorrelation function and general grows over time without bound. This indicates that flicker noise cannot be considered cyclostationary and must be modelled differently from thermal and shot noise.

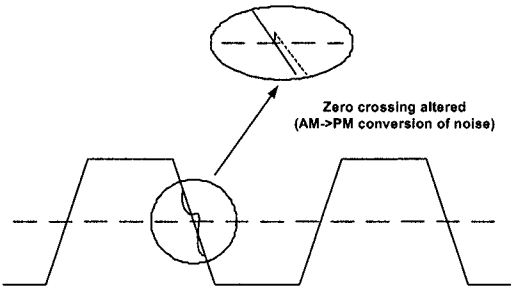


Figure 7-8. Output of a DFD with noise injected

A method of characterizing jitter in a cyclostationary system is the impulse sensitivity function (ISF) [7]. The ISF assumes that the system is a time-varying linear system. The ISF is a unitless time-varying function and can be expressed as

$$T_j(t) = \Gamma(t) \cdot n(t) \tag{7.18}$$

where $T_j(t)$ is the jitter, n(t) is the amplitude noise, and $\Gamma(t)$ is the ISF. For small noise sources, the relationship between amplitude noise and jitter in DFDs can be considered to be fairly linear. A typical ISF function and its corresponding DFD output waveform are shown in Figure 7-9.

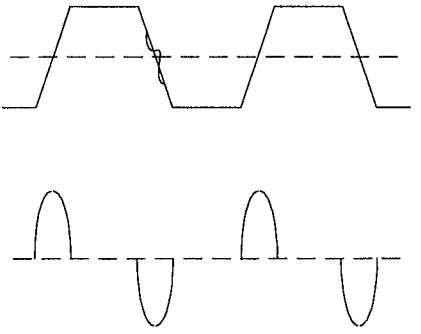


Figure 7-9. ISF of a DFD output waveform

There are many source of jitter in DFDs. The three primary sources are power supply noise, substrate noise, and device noise. Power supply noise usually comes in the form of correlated noise, since it is common to all components in the frequency divider. Substrate noise has become less important due to recent enhancements in power routing strategies [8]. Therefore, in this section, only power supply noise and device noise is considered.

7.3.3 Synchronous Divider Model

A generic synchronous divider is shown in Figure 7-10. As the figure shows, any frequency divider architecture can be chosen to perform the exact frequency division. Once the division is performed it is re-synchronized with the input clock by using a retiming flip-flop. In order to guarantee correct operation, the divider architecture must be chosen such that the output is not produced during an input clock edge transition. Violating this timing may result in a race condition between the data and clock terminals of the retiming flip-flop and cause frequency division errors.

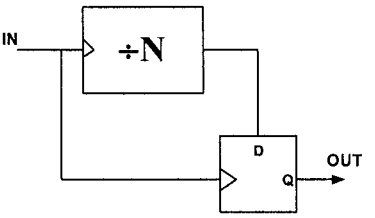


Figure 7-10. Synchronous Divider

In this architecture, the jitter can be given as

$$T_j(t) = T_{j,clk} + T_{j,FF}(t) \tag{7.19}$$

where $T_{j,clk}$ is the clock jitter and $T_{j,FF}(t)$ is the time-varying jitter induced by the retiming flip-flop. It is clear from equation (7.19) that the main advantage of synchronous dividers is the elimination of jitter that may have been induced by the frequency divider core. The noise of the input clock signal and the dividers are usually not correlated. Assuming white input noise, the output jitter variance can be given as

$$\sigma_j^2(t) = \sigma_{j,clk}^2 + \sigma_{j,FF}^2(t) \tag{7.20}$$

where $\sigma_{j,clk}^2$ is the variance of the clock jitter, $\sigma_{j,FF}^2(t)$ is the time-varying variance of the flip-flop jitter.

7.3.4 Asynchronous Divider Model

A generic asynchronous divider model is shown in Figure 7-11. As the figure shows, it consists of identical cascaded divider units. The output of each stage drives the clock input terminal of the next stage. The final output of the divider is typically taken from the last stage. The classical model of such as divider is given as

$$T_j = T_{j,clk} + \sum_{i=1}^{N} T_{j,i} \tag{7.21}$$

where N is the number of divider units, $T_{j,clk}$ is the input clock jitter, and $T_{j,i}$ is the jitter induced by the i^{th} divider unit. Assming that the noise in the divider stages is uncorrelated (thermal and shot device noise), the output jitter variance can be given as

$$\sigma_j^2(t) = \sigma_{j,clk}^2 + N \cdot \sigma_{j,div}^2 \tag{7.22}$$

where $\sigma_{j,div}^2$ is the variance of the jitter in a divider unit. It is assumed here that all divider units are identical and have the same variance of jitter. If jitter in the divider stages is correlated (in the case of flicker noise or supply noise), the jitter expression becomes

$$\sigma_j^2(t) = \sigma_{j,clk}^2 + \left(N \cdot \sigma_{j,div}\right)^2 \tag{7.23}$$

Comparing (7.22) and (7.23) to (7.19), it is clear that the jitter of asynchronous divider is much higher than that of synchronous dividers, especially in the case of correlated noise.

This classical analysis, however, ignore the time-varying nature of jitter in DFDs. Rewriting (7.21) with time-varying nature taken into account, results in the following expression

$$T_j(t) = T_{j,clk} + \sum_{i=1}^{N} T_{j,i}\left(t - \left(N - i \cdot \Delta T\right)\right) \tag{7.24}$$

where ΔT is the delay in a divider unit.

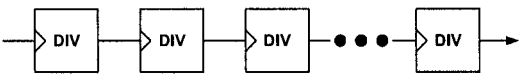


Figure 7-11. Asynchronous Divider

It is interesting to take two boundary conditions into account. As $\Delta T \rightarrow 0$, the variation in jitter at different noise injection times becomes quite severe. In other words, jitter becomes very sensitive to its time-varying nature. For

example, if noise injection only occurs when the impulse sensitivity function of the divider is zero, then

$$T_j(t)\Big|_{ISF=0,\Delta T\to 0} = T_{j,clk} \tag{7.25}$$

This means that the divider's output jitter is only equal to that of the clock input! The asynchronous DFD performance is at least as good as that of the synchronous divider. When noise is injected when the ISF has a maximum value, then the jitter is given as

$$T_j(t)\Big|_{ISF=max,\Delta T\to 0} = T_{j,clk} + N \cdot T_{j,div} \tag{7.26}$$

This shows that there is a large variation in jitter depending on when the jitter is applied.

In the other extreme case, when $\Delta T \to T_{CLK}$ (the input clock period), there are two general expressions for jitter, depending on the nature of the injected noise. In the case of correlated noise (flicker noise, supply noise), the jitter expression becomes

$$T_j(t) = T_{j,clk} + n_j \cdot \sum_{i=1}^{N} \Gamma\left(t - \left(N - i \cdot T_{CLK}\right)\right) \tag{7.27}$$

which simplifies to

$$T_j(t) = T_{j,clk} + n_j \cdot \Gamma_{avg} \tag{7.28}$$

For a symmetric ISF, the average value is zero. This means that in this case, the jitter of the asynchronous divider from a correlated noise source is minimum. For the other case of when $\Delta T \to T_{CLK}$ and the noise source is uncorrelated, then the noise adds in a root mean square fashion.

7.3.5 Results

In this section, a few experiments on jitter in frequency divider are described. The first experiment consists of adding an undesired frequency tone to the input clock and measuring the jitter though a divider. Figure 7-12 shows the resulting jitter. As shown, the resulting jitter varies with the location of the frequency spur. As the spur frequency approaches the divider output frequency, the total jitter increases. This means that the jitter depends on both N_{div}, the frequency division ratio, as well as T_{CLK}. It was also found that the jitter increases when the period of the applied spur approached $j \cdot N_{div} \cdot T_{CLK}$, where j is an integer. The implication of this observation is that magnitude of the spur does not always decrease by 6dB for each doubling in frequency division ratio. As the figure shows, in some cases, the spur strength may actually increase.

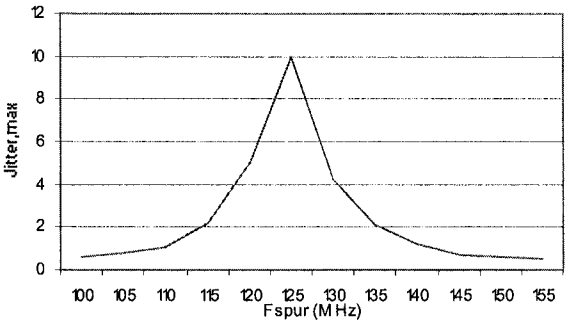


Figure 7-12. Jitter versus input frequency spur location

Another experiment was conducted in order to verify the asynchronous divider model. The delay in the divider stages was varied with correlated noise added to the supply voltage and the resulting jitter was measured. A static CMOS divider was used, meaning that the ISF is symmetric (zero mean). The result of this experiment is shown in Figure 7-13. Two cases are shown. In one case, jitter was added when the ISF of the first divider was minimum. In this case, the jitter was small for small divider delay. The jitter gradually grows, but then decreases again as the delay approaches the input clock period. This is consistent with equations (7.25), (7.26), and (7.28).

When jitter was added when the ISF of the first divider was maximum, the total jitter is very high with very small divider delays. As the divider delay increases, the jitter decreases almost linearly and approaches the jitter of just the input clock. This too is consistent with what equation (7.25), (7.26), and (7.28) predict.

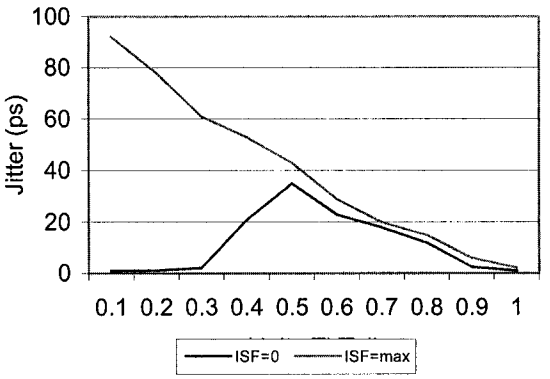


Figure 7-13. Jitter versus divider unit delay

7.4 Fractional-N PLLs as Clock Generators

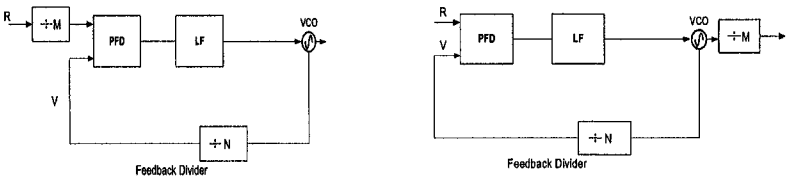
SoC processors often embed a heterogeneous mixture of signal processing blocks, each requiring a specific sampling rate. Often, a fixed reference crystal is provided to generate the required frequencies. One class of methods of providing the required frequency is by using one of the structures shown in Figures 7-14(a) and 7-14(b). In Figure 7-15(a), the input crystal oscillator is pre-divided by M and the feedback divider value is N. The output frequency is given as

$$f_{out} = \frac{N}{M} f_{in} \tag{7.29}$$

If the value of M is large, this would significantly reduce the loop bandwidth of the PLL, requiring a large loop filter (large area) and excessively long lock time. Another solution is to use the structure shown in Figure 7-15(b). In this case, the PLL is followed by a divide-by-M post-divider. The output frequency is also given as

$$f_{out} \frac{N}{M} f_{in} \tag{7.30}$$

If the post-divider ration, M, is large, the VCO would be required to work at a high frequency. More specifically, the VCO in this clock generator configuration must work at M times the VCO of the configuration of Figure 7-15(a). To illustrate this point, consider the following example. If the input reference frequency is 13MHz and the desired output frequency is 59MHz, then N=59, M=13. The VCO frequency, in this case, will operate at 767MHz. If an additional frequency of 33MHz is required, then a second frequency divider of will be needed. The VCO will have to operate at 13MHz times the least common multiple of the two desired output frequencies. This means that VCO frequency will be well above 10GHz! For low-power portable systems, the power consumed by this clock generator configuration may be too excessive.



(a) (b)

Figure 7-14. Clock Generator topology employing (a) pre-divider and (b) post-divider

7.4.1 Basic Toplogy

One method to remedy the above stated problem is to use a fractional-N PLL architecture [18]. The frequency divider in fractional-N PLLs toggles between one of two division ratios to generate an average division ratio, N.f, that lies between these two division ratios, where N is the integer division ratio and f is the fractional division ratio. The division ratios are usually chosen such that they differ by a factor of one. For example, if a division ratio of 4 is chosen in four clock cycles and a division ratio of 5 is chosen in one clock cycle, an average division ratio of (4*4+5*1)/5=4.2 is achieved.

A diagram of a fractional-N PLL is shown in Figure 7-15. As the figure shows, an accumulator is required to count by an input fractional step. The "carry out" bit of the accumulator is used as a modulus control on the frequency divider.

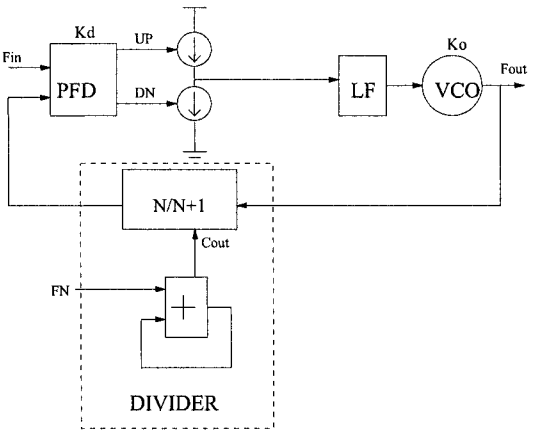


Figure 7-15. Fractional-N Phase-locked Loop

To illustrate the advantage of fractional-N PLL for clock generators, consider the previous example. In this case, the VCO frequency can be chosen independently from the reference frequency. Therefore if only one frequency of 59MHz is desired, M=2, N=9+1/13, and the VCO frequency is only 118MHz. In the second scenario when two frequencies are desired, the VCO only needs to operate at 1947MHz. Although this is still a bit high, it is an order of magnitude improvement over an integer PLL approach.

The main disadvantage of fractional-N PLLs is the presence of fractional spurs. When used as sampling clock for a sigma-delta data converter, the fractional spurs can serious degrade the data converter's SNR, as explained in section 7.2.

7.4.2 Sigma-Delta PLLs

7.4.2.1 Basic Operation

One PLL architecture that reduces fractional spurs, which has recently gained popularity, is known as *sigma-delta compensated fractional-N PLL*, or simply ΔΣ PLL [19-22]. Instead of attempting to eliminate the phase errors in fractional-N PLLs, this technique noise shapes the phase errors to higher frequencies. This is done by using the output bits of a digital ΔΣ modulator to change the frequency division ratio in a random manner. The closed loop bandwidth then acts as a low-pass filter, which eliminates the noise shaped error terms. Figure 7-16 shows a block diagram of a ΔΣ PLL. Note that the sigma-delta modulator is an entirely digital block, clocked by the output of the divider signal. The major advantages of this technique include larger loop bandwidths (which result in faster lock times), less phase noise due to noise in the PFD and charge pumps. The two major disadvantages of this technique are increased circuit complexity (which entails larger power dissipation and area) and high phase noise at higher frequency offsets.

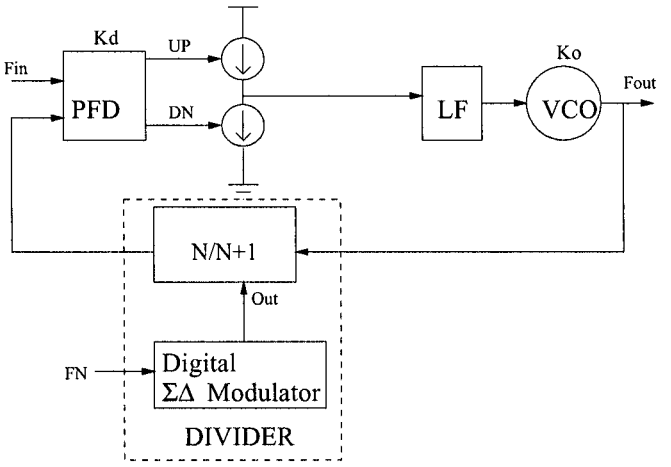


Figure 7-16. ΣΔ PLL Architecture

The frequency shaping property of ΣΔ modulators is best understood in terms of its transfer function. A model of a conventional first order ΣΔ modulator is shown in Figure 7-16. Note that the quantization noise is modeled as additive noise. This model is only justified if the input to the modulator has sufficient activity [2]. Under this assumption, the quantization noise will be uniformly distributed with $\sigma^2=1/12$. The transfer function of this feedback loop is

$$Y(z) = \frac{1/(1-z^{-1})}{1+z^{-1}/(1-z^{-1})} G(z) + \frac{E_q(z)}{1+z^{-1}/(1-z^{-1})} \tag{7.31}$$

which may be simplified to

$$Y(z) = G(z) + (1-z^{-1})E_q(z) \tag{7.32}$$

As Equation (7.32) indicates, the input signal passes unchanged and the quantization noise is shaped by the $(1\text{-}z^{-1})$ function, which is a high-pass filter.

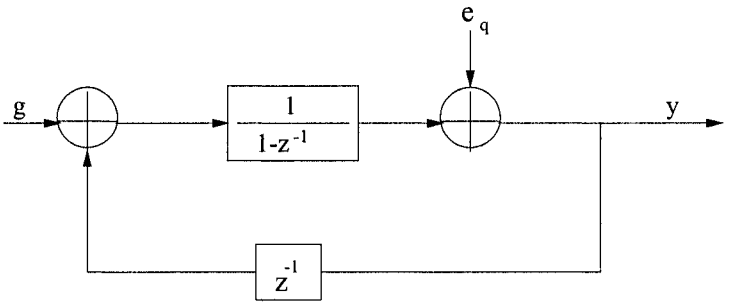


Figure 7-17. First order ΣΔ modulator model

One way of achieving even less quantization noise at higher offset frequencies is to cascade several first order ΣΔ modulators, as shown in Figure 7-18. This type of architecture is called a MASH ΣΔ modulator [2]. Since there are three first-order ΣΔ modulators (each accumulator is a first order sigma-delta modulator), this is a third order MASH ΣΔ modulator. The sum term of the accumulator is the error term output of the sigma-delta modulator. The transfer functions of each modulator section is given as

$$Y_1(z) = G(z) + (1-z^{-1})E_{q1}(z) \tag{7.33a}$$
$$Y_2(z) = -E_{q1}(z) + (1-z^{-1})E_{q2}(z) \tag{7.33b}$$
$$Y_3(z) = -E_{q2}(z) + (1-z^{-1})E_{q3}(z) \tag{7.33c}$$

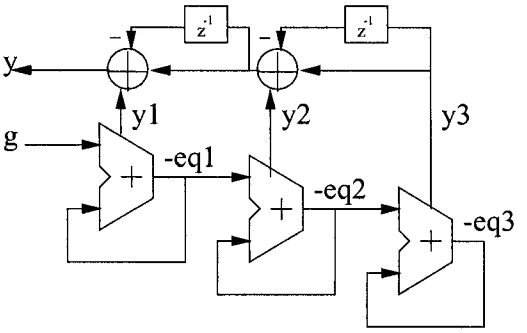


Figure 7-18. Third order MASH ΣΔ digital modulator

The modulator's total output is given as

$$Y(z) = Y_1(z) + (1 - z^{-1})\left[(1 - z^{-1})Y_3(z) + Y_2(z)\right] \tag{7.34}$$

which can be simplified as

$$Y(z) = G(z) + (1 - z^{-1})^3 E_{q3}(z) \tag{7.35}$$

In general, for n^{th} order MASH ΣΔ modulators, the output of the modulator becomes

$$Y(z) = G(z) + (1 - z^{-1})^n E_{qn}(z) \tag{7.36}$$

where E_{qn} is the quantization noise of the n^{th} sigma delta modulator. In the context of fractional-N PLLs, the quantization error has a variance of $\delta^2/12$ over a bandwidth of F_{ref}, where F_{ref} is the comparison frequency at the PFD and δ is the step size. Assuming division ratio varies between N and N+1, the step size (δ) is equal to one. Consequently, the power spectral density of the quantization noise is $1/(12F_{ref})$. The power spectral density of the modulation noise (or frequency fluctuation) then becomes

$$N(f) = E_q(f) \cdot \left|(1 - z^{-1})^n F_{ref}\right|^2 = \left|1 - z^{-1}\right|^{2n} \cdot (F_{ref}/12) \tag{7.37}$$

where n is the order of the modulator and $E_q(f)$ is the power spectral density of the quantization noise . Since we are interested in the phase noise contribution of the ΣΔ modulator, not the frequency noise, Equation (7.37) must be integrated. Integrating in the z-domain amounts to multiplying by $\frac{1}{1-z^{-1}}$; therefore, the power spectral density of the phase fluctuations becomes

$$N_\phi(f) = \left(\frac{2\pi T_{ref}}{1-z^{-1}}\right)^2 \cdot \left(\frac{(1-z^{-1})^{2n} F_{ref}}{12}\right) = \frac{(2\pi)^2}{12F_{ref}}\left|1-z^{-1}\right|^{2(n-1)} \quad (7.38)$$

Converting this from the z-domain to the frequency domain gives

$$N_\phi(f) = \frac{(2\pi)^2}{12F_{ref}}\left[2\sin\left(\frac{\pi f}{F_{ref}}\right)\right]^{2(n-1)} \quad (7.39)$$

One important issue is to note the exponent term on the sine function in Equation (7.39). In an analog ΣΔ modulator, this term would simply be "2n". The input to a digital ΣΔ modulator, however, is a "frequency" control word. This frequency control word must first be integrated to produce a phase. In the digital domain, an integrator may be realized by an accumulator. Note, however, that an accumulator would add a pole, or factor of $1/(1-z^{-1})$ in the digital domain to the ΣΔ transfer function (Equation 7.32). This has the effect of reducing the order of the ΣΔ modulator by one.

One disadvantage of using a higher order modulator is increased hardware complexity. Fig. 7-19 shows the spectral densities of quantization noise shaped by second and third order modulators. As the figure shows, using a higher order ΣΔ modulator results in less quantization noise at lower frequencies; however, the noise rises faster and becomes more difficult to suppress at frequencies near $F_{ref}/2$.

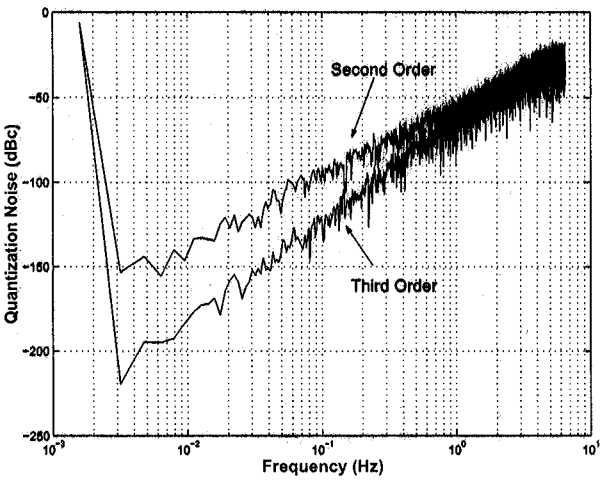


Figure 7-19. Spectral densities of second and third order modulators

7.4.2.2 Sigma-Delta Dithering

The above analysis assumes that the quantization error is a uniformly distributed random process. In general, if a DC signal is applied to a sigma-delta modulator, a tonal output response may result. This is especially true for a MASH sigma-delta modulator. In order to randomize the output a dithering function is required. Dithering can generated by using a linear feedback shift register (LFSR) that is maximally encoded [1]. An LFSR will generate a random sequence of 1's and 0's with a bit sequence length of 2^L-1, where L is the number of registers used in the LFSR. The output of the LFSR is usually added to the lower LSBs of the input of the sigma-delta modulator.

7.5 Oversampled PLL Topologies

In practical SoC processors, more than one signal processing engine is integrated on-chip. Some engines may be digital signal processing hardmacros, others may be sampled analog systems, or data converters. Such sub-systems require specific clock rates that are completely unrelated. The simplest method of providing a clock for each sub-system is to have a dedicated fractional-N or integer PLL for each sub-system. This method can prove both power hungry and area inefficient.

One possible remedy for the above stated problem is to have a single PLL working at the least common multiple of all the required frequencies. The required frequencies can then be easily obtained by integer dividing down the PLL's output frequency. The clock generation architecture is shown in Figure 7-20. A fractional-N PLL can be very useful in such an architecture in that the least common multiple frequency is no longer constrained by the PLL's input reference frequency.

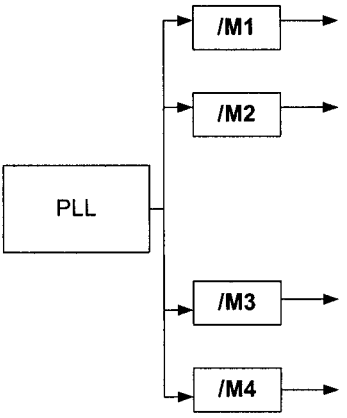


Figure 7-20. Clock generation architecture using one PLL

When used as a sampling clock for an analog system or a data converter, the spurs generated by the fractional-N PLL must be taken into account. Fractional spurs can effectively lower the dynamic range of data converters and must be taken into account. One method of reducing the magnitude of the fractional spurs is to increase the PLL's output frequency and divide it by a larger value. This is similar to the oversampling technique used in data converters. The fractional spurs are effectively lowered by $20 \cdot \log_{10} M$, where M is the value of the post-divider. Hence, for every doubling of the PLL's output frequency, the spur levels are reduced by 6dB. Therefore, for oversampled PLL architectures, there is a direct trade-off between power consumption and spectral purity. Alternatively, fractional spurs can be eliminated by using the sigma-delta PLL, as explained in the previous section.

7.6 Direct Digital Synthesis with Analog Interpolation

Modern SoC processors integrate both analog and digital real-time functions such as audio and video. Such applications often require a clock generator to produce several unrelated frequencies for digital signal processing as well as sampled analog sub-systems. One solution is to create one phase-locked loop (PLL) clock generator running at a high frequency that can then be integer divided down to obtain all desired frequencies [25]. The disadvantage of this approach is high power consumption and stringent jitter requirements on the PLL. Another method is to have a dedicated PLL for each clock domain. This solution is very costly in terms of power and area.

One solution would be to use several accumulators to produce the desired clock signals from one PLL frequency. The disadvantage of using an accumulator is that the frequency resolution is limited to the number of bits of the accumulator. Alternatively, an MN counter (Figure 7-21) can be used in place of an accumulator, giving an output frequency precisely equal to M/N times the input frequency. Another disadvantage is that the worst case jitter produced by MN counter is equal to an entire input period, which for many applications, is too high. Moreover, the periodic nature of the MN counter results in frequency tones that are often too high for practical use as a sampling clock for analog sub-systems.

One way to further reduce the jitter produced by MN counters is to interpolate the correct phase of the output signal using analog techniques. This has been previously attempted in [23] and [24]. In both [23] and [24], the contents of the accumulator is converted to analog form by digital-to-analog converter (DAC). The analog value is subtracted from the previously value of the accumulator. The difference is detected and integrated. Once

the integrated analog value reaches a certain level, the output of the last stage comparator is triggered. The output of the comparator is then fed through a toggle flip-flop (TFF), which gives the final output.

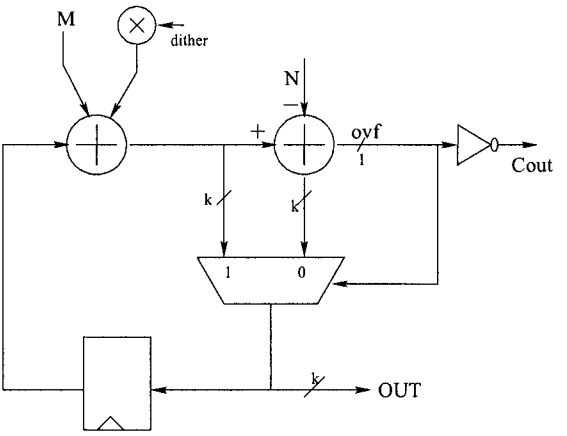


Figure 7-21. MN Counter Block Diagram

The class of implementations discussed above rely on absolute reference levels for threshold triggering and are prone to offset errors. In this section, an alternate method is discussed that achieves both low-power and low-jitter [30]. An MN counter with analog phase interplation (MNA) is used. The techniques used include fully differential and an innovative feedback controlled replica bias circuit topologies to accurately produce a reference current to produce an accurate phase delay across all process and temperature corners. Moreover, this technique allows for drastic reduction of capacitor and current sizes, reducing area and power consumption.

7.6.1 MNA Concept and Architecture

An MN counter with analog interpolation (MNA) is used to further reduce jitter. The basic concept behind the MNA counter is best illustrated through an example, shown in Figure 7-22. A modulo-8 counter is assumed with an input of 3. This means that the input frequency, F_{CLK}, is divided by 3/8. The output of the counter is shown in the top row. The average output period of the MN counter is $\frac{8}{3}\cdot\frac{1}{F_{CLK}}$. The instantaneous period, however, can be $\frac{2}{F_{CLK}}$ or $\frac{3}{F_{CLK}}$, giving a maximum cycle-to-cycle jitter of $\frac{1}{F_{CLK}}$.

The MNA counter attempts to reduce the jitter by shifting the position of the rising edge of the clock such that all clock periods are equal to one another. This is done by advancing the rising edge by a value given by

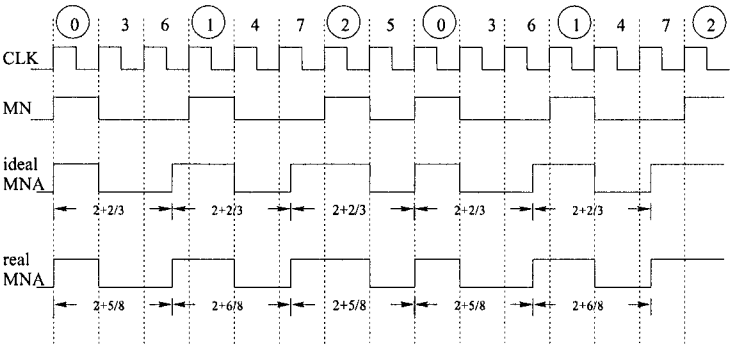


Figure 7-22. MNA Example with M/N=3/8

$$\Delta T = \frac{ACC_i}{3} \tag{7.40}$$

where ACC_i is the contents of the accumulator at the time of overflow. As the row labelled "ideal MNA" in Figure 7-22 shows, each clock cycle period is exactly $\left(2+\frac{2}{3}\right)\cdot\frac{1}{F_{REF}}$. If the circuitry used to produce this delay is perfect, zero jitter is introduced by the MNA counter. In practice, there is a trade-off between the accuracy of the delay generator and the area and power consumed by the MNA counter. The last row in Fig. 2 shows the output of an MNA counter with a delay generator having 3-bit accuracy. In this case, a fraction of $\frac{1}{3}$ is estimated as $\frac{3}{8}$, and a fraction of $\frac{2}{3}$ is estimated as $\frac{5}{8}$. Using these estimates, the output period varies from $\frac{21}{8}\cdot\frac{1}{F_{REF}}$ to $\frac{22}{8}\cdot\frac{1}{F_{REF}}$, giving a cycle-to-cycle jitter of $\frac{1}{8}\cdot\frac{1}{F_{REF}}$. In general, for an n-bit accuracy delay generator, the maximum cycle-to-cycle jitter will be $\frac{1}{2^n}\cdot\frac{1}{F_{REF}}$.

Figure 7-23 shows a block diagram of the proposed MNA counter. An inverse operator is needed to compute the 1/M value. An integer multiplier is required to produce the ACC*(1/M) term. A negative delay is produced by the delay generator by operating it one cycle early (using the eovf signal) and delayed by $\left(2^n - \frac{ACC}{M}\right)\cdot\frac{1}{F_{REF}}$. This operation is implemented by taking the one's complement of the multiplier's output.

A schematic diagram of the delay generator is shown in Figure 7-24. The basic concept of the delay generator is to generate a discrete number of delays by charging and discharging a bank of capacitors, given a digital word controlling the switches on the capacitors. The delay generated is given as

$$T_{delay} = \frac{C_L V_{swing}}{I} \tag{7.41}$$

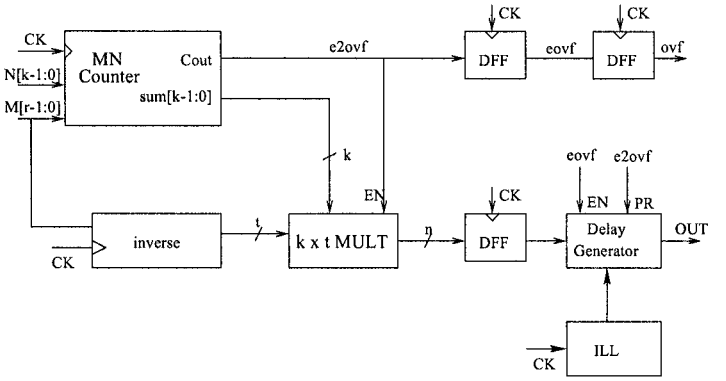


Figure 7-23. MNA Counter Block Diagram

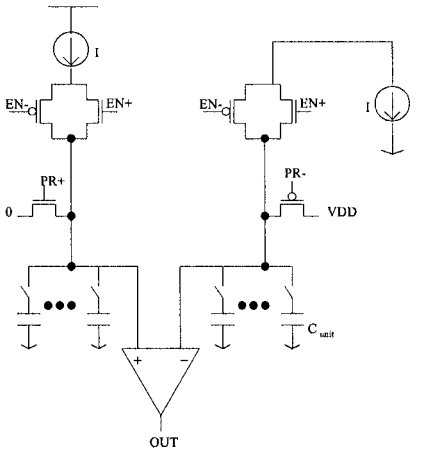


Figure 7-24. Schematic Diagram of Delay Generator

where I is a constant current input, V_{swing} is the voltage swing of the delay generator, and C_L is the load capacitance, which is dominated by the bank of capacitors in the delay generator. Instead of relying on a constant voltage reference, a pair of bank of capacitors is used: one is initially charged to VDD and the other is initially discharged to 0V. When the delay generator is enabled, one node begins charging to VDD and the other discharges towards 0V. An analog comparator compares the two voltages. When they are equal to one another, an output edge is produced. In this case, the delay generated is equal to

$$T_{delay,i} = \frac{C_{unit} \cdot ACC_i \cdot V_{DD}}{M \cdot I \cdot 2} \tag{7.42}$$

where C_{unit} is the unit capacitance in the bank of binary weighted capacitors. The number of binary weighted capacitors are chosen according to the amount of jitter reduction is required. For example, if a factor of 16 reduction of jitter is required, 4 binary weighted capacitors are needed.

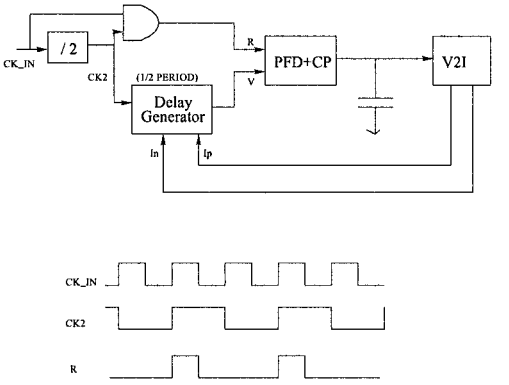


Figure 7-25. ILL (a) schematic and (b) timing diagrams

One major concern with the delay generator is the variation of its performance with process, and most notably the applied current. This may be remedied by using using a feedback replica bias technique such that temperature and process are tracked. Such a circuit results in a current locked loop (ILL). An ILL schematic diagram is shown in Figure 7-25(a). Figure 7-25(b) shows the timing of the critical nodes.

The reference signal is the output of the PLL divided by two and delay by half a period. The analog delay generator is fixed to generate a one-half a period delay when the ILL is locked. The phase-frequency detector (PFD) compares the delay generator and the reference signal phases. After low-pass filtering, the the phase error is eventually translated into a current, feeding into the replica delay generator.

The closed loop transfer function can be expressed as

$$H(s) = \frac{\frac{I_p}{2\pi} \cdot \frac{1}{C} \cdot K_{v2i}}{s + \frac{I_p}{2\pi} \cdot \frac{1}{C} \cdot K_{v2i} \cdot K_{TD}} \tag{7.43}$$

where I_p is the charge pump current, C is the loop filter capacitor, K_{v2i} is the gain of the V2I converter (uA/V), and K_{TD} is the gain of the delay generator (ns/uA). This is a single pole system that is unconditionally stable. The ILL, however, can false lock to zero or double input period. False locks can be prevented by careful control of the forward gain of the ILL. To save

power consumption, the ILL can be turned off when phase interpolation is not needed, then back on 1 or 2 cycles in advance.

7.6.2 Parameter Analysis of MNA Counter

In order to demonstrate the MNA's dynamics as a function of its parameters, a 6-bit MNA counter is considered (for 32x jitter reduction). A critical parameter in the MNA counter is the choice of the unit capacitor size. The capacitor size directly affects jitter, area, and current consumption. Figure 7-26 shows the effect of unit capacitor size on jitter reduction of the MNA counter. As the figure shows, the jitter reduction capability of the MNA counter degrades with increasing capacitor size, due to increased nonlinearity of the current sources at higher current levels. For very small capacitor sizes, however, parasitic capacitance becomes comparable to the unit capacitor size, which degrades the performance of the MNA. For this reason, a C_{unit} size of 30fF gives best performance.

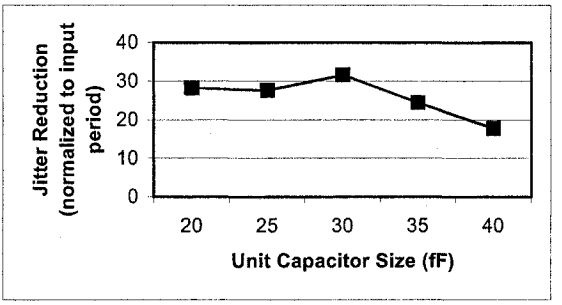


Figure 7-26. C_{unit} size versus jitter reduction factor

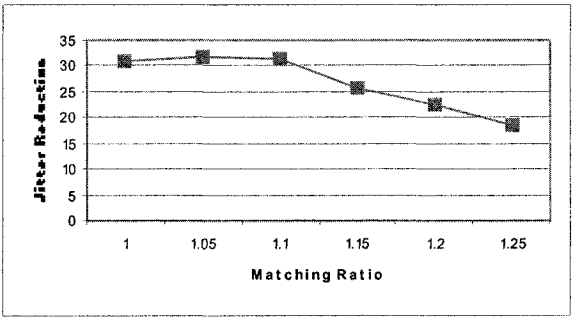


Figure 7-27. Effect of mismatch in complementary sides of the delay generator

Another concern in using small capacitor sizes is the increased random mismatches between capacitor elements. Figure 7-27 shows the effect of

mismatch between complementary ends of a delay generator on jitter reduction factor. As the figure shows, good performance is maintained for up to 10% mismatch. Similar effect is true for the effect of mismatch between the delay generator and replica delay generator capacitor arrays. This clearly demonstrates the stability of the MNA counter's performance due to its differential nature and the used of an ILL.

7.6.3 Circuit Implementation

The input stage of the MNA counter consists of a high-speed MN counter. To speed up arithmetic a high-speed 12-bit 2-stage carry select adder [26]. This has shown to give good performance with low area penalty. Using this approach, the critical path is reduced to a total of 5 full-adders (FA), 3 multiplexers, and a D-type flip-flop setup delay.

Although its activity is low, the multiplier execution window is equal to the high input clock rate. For this reason, a 3-stage pipelined multiplier is used, as shown in Figure 7-28. The first stage is a latch, inhibiting the input from the MN counter to ripple through the multiplier if an overflow signal is not generated. The first pipeline stage consists of the partial product generation and accumulation tree. The next two stages is a pipeline final stage carry select adder. The size of the multiplier is 12-bit × 6-bit integer multiplier.

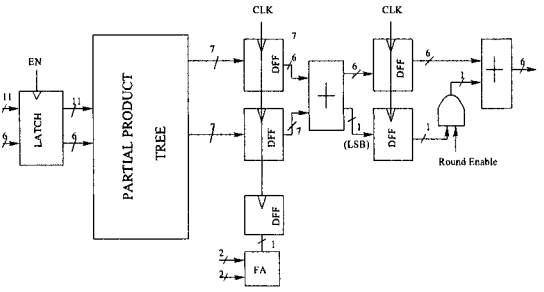


Figure 7-28. 12x6 bit Pipelined Integer Multiplier

The inverse operator produces a digital value equal to 1/M. This is implemented by using a serial division algorithm [26]. The output of the divider is rounded to reduce truncation error by 0.5LSB.

The 3-state PFD [28] is used. A current steering charge pump is used to reduce the charge pump's settling time. Wide-swing cascode mirrors [28] have been used to ensure that the UP and DN current sources remain in saturation and are well matched. The loop filter is implemented as a 1.4pF MOS cap. Wide-swing cascode mirrors have been used in the V2I converter. For best matching, the current mirrors for both the replica delay

generator and the delay generator were implemented as one unit using common centroid layout [28].

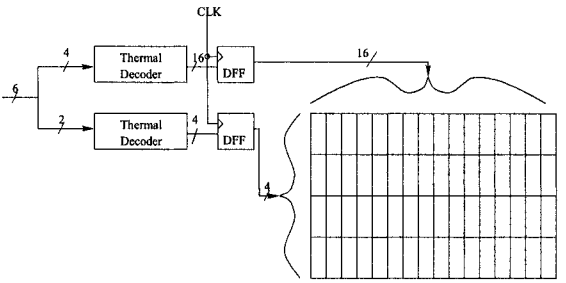


Figure 7-29. Conceptual Diagram of the capacitor based DAC

Conceptually, the capacitor based DAC is implemented as shown in Figure 7-29. A thermal decoder decodes the input signal into row and column address lines. The DFF registers shown ensure glitch free transition on the DAC. The array is configured as a 16-by-4 array DAC. It consists of local decoding logic, a switch, and a MOS capacitor.

Matching in the capacitors used in the replica delay generator and the delay generator is also very important. For best matching, the capacitors of both the replica delay generator and the delay generator are implemented as one array arranged in a two-dimensional common centroid layout [29]. The dimensions of this array are 128-by-10 array cells. Worst case matching between two capacitors is 0.6%. In comparison, [23] and [24] have programmable current sources, which tend to be much larger to achieve the same accuracy as in this work.

7.6.4 Results

Table 7-1 provides a performance summary for the MNA counter. Low-area has primarily been made possible by switching capacitors instead of current. As Figure 7-30 shows, the area of the capacitor array is comparable to that of the dense digital logic. Low-power is attributed to small size of unit capacitor size and shutdown of the ILL between output toggles. High accuracy is also attributed to a differential architecture and the use of an ILL.

Table 7-2 gives a summary of the comparison of the proposed technique with the two other state-of-the-art techniques [23-25]. The figure of merit used for comparison is:

$$FOM = \underbrace{\frac{1}{2^n}\left(\frac{DigitalPwr}{Fin} + \frac{Ana\log Pwr}{Fout}\right)}_{power-consumption} \Big/ \underbrace{\left(20\cdot\log\left(2^n\right) - spur_{reduction}(dB)\right)}_{jitter} \cdot \frac{Area}{2^n} \tag{7.44}$$

where n is the digital bitwidth. The power consumption portion of the figure of merit is divided by 2^n since the power consumption of both the analog and digital sections increase with n exponentially. Area is also taken into account.

Table 7-1: MNA performance summary

Technology	0.25um CMOS
Power Supply	2.5V
Power Consumption	1.5mA @ 100Mhz
SFDR	37dB
Lock Time	<1usec
Area	0.12mm^2
Max cycle-to-cycle jitter	280ps @ 125MHz

Table 7-2. Proposed versus state-of-the-art

	Proposed	**[23]**	**[24]**	**[27]**
Area (mm^2)	0.12	N/A	12	4.5
Power (mW/MHz)	0.0375	5.63	0.31	0.22
Spur Reduction	30dB	40dB	40dB	40dB
Digital bitwidth	6	8	8	8
FOM (normalized)	1	40.29	2.23	

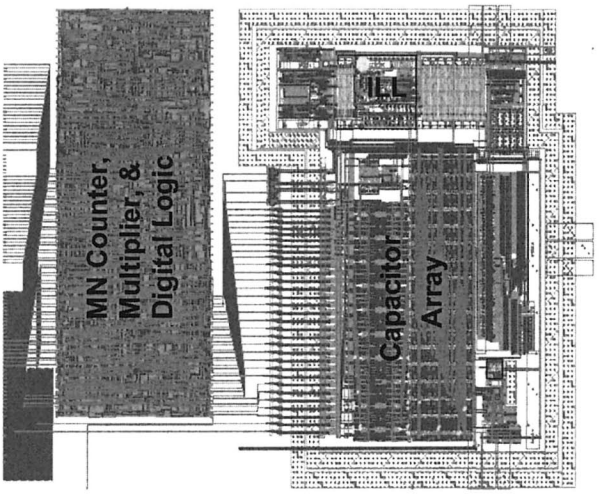


Figure 7-30. Layout Plot of MNA Counter

7.7 Summary

In this chapter, clock generation requirements and techniques for digital signal processing (DSP) and data converters have been reviewed. The requirements of DSP blocks amount to reducing the cycle jitter in order to maximize the MACs per second rating. The requirements for data converters depend on the type of data converters and what phase noise spectrum the clock generator has. For Nyquist rate data converters, the requirements for data converters are straightforward. For oversampled sigma-delta ADCs, the requirements are more complicated. As it turns out, data converters, especially sigma-delta converters, are more sensitive to deterministic jitter, or spurs, than to white Gaussian noise.

Frequency dividers inserted in the clock tree after the PLL can also contribute to the overall jitter. The jitter in frequency dividers has also been analyzed. The time-varying nature of digital frequency dividers has been demonstrated. Asynchronous dividers can be designed to have low jitter by manipulating the ISF and when the noise is injected into the dividers.

There are many methods to design clock generation circuitry for DSP and data converters. The simplest method is to use a dedicated PLL for each clock frequency required. This, however, can be expensive in terms of area and power consumption. A smarter solution is to use a fractional-N PLL followed by a bank of frequency dividers. A fractional-N PLL is an attractive solution for multi-mode systems where several reference frequencies need to be supported. The PLL output then operates at the least common multiple frequency of all required frequencies. This frequency, however, can be very high and can have large power consumption and make the PLL more difficult to design. An alternative solution is to have a bank of direct digital synthesizers with analog interpolation instead of frequency dividers. This allows more flexibility in the choice of the PLL output frequency and an integer-PLL can be supported.

REFERENCES

[1] A. Oppenheim and R. Schafer, *Discrete-time Signal Processing*, Englewood Cliffs, New Jersey: Prentice Hall, 1989.

[2] S. Norsworthy, R. Schreier, and G. Temes, *Delta-Sigma Data Converters*, New York: IEEE Press, 1997.

[3] J. Candy and G. Temes, *Oversampled Delta-Sigma Data Converters*, New York: IEEE Press, 1992.

[4] N. Dalt, et. al., "On the Jitter Requirements of the Sampling Clock for Analog-to-Digital Converters," *IEEE Trans. Circuits and Systems I*, vol. 49, no. 9, pp. 1354-1360, September 2002.

[5] S. Haykin, *Digital Communications*, Wiley: New York, 1988.

[6] B. Razavi, *Design of Analog CMOS Integrated Circuits*, Boston: McGraw Hill, 2001.

[7] A. Hajimiri and T. Lee, "A General Theory of Phase Noise in Electrical Oscillators," *IEEE J. of Solid-State Circuits*, vol. 3, no. 2, pp. 179-194, February 1998.

[8] P. Larsson, "Measurements and Analysis of PLL Jitter Caused by Digital Switching Noise," *IEEE J. of Solid-State Circuits*, vol. 36, no. 7, pp. 1113-1119, July 2001.

[9] V. Kroupa, "Jitter and phase noise in frequency dividers," *IEEE Transactions on Instrumentation and Measurement*, vol. 50, Oct. 2001, pp. 1241-1243.

[10] J. Crockett and M. Siccardi, "Power supply noise conversion to phase noise in CMOS frequency digital divider," *IEEE Int'l Frequency Control Symposium*, May 2002, pp. 699-702.

[11] T. Yamaguchi, et. al., "Timing jitter measurement of 10 Gbps bit clock signals using frequency division," *IEEE VLSI Test Symposium*, May 2002, pp. 207-212.

[12] S. Levantino, et. al., "Phase noise in digital frequency dividers," *IEEE J. of Solid-State Circuits*, vol. 39, no. 5, May 2004, pp. 775-784.

[13] O. Llopis, et. al., "Phase noise performance of microwave analog frequency dividers: application to the characterization of oscillators up to the millimetre-wave range," *IEEE Trans. On Ultrasonics, Ferroelectrics and Frequency Control,* vol. 46, no. 4, July 1999, pp. 935-940.

[14] E. Rubiola, et. al., "Phase noise in regenerative frequency dividers," *IEEE Trans. On Instrumentation and Measurement*, vol. 41, no. 3, June 1992, pp. 353-360.

[15] M. Driscoll and T. Merrell, "Spectral performance of frequency multipliers and dividers," *IEEE Proc. Of Frequency Control Symposium*, May 1992, pp. 193-200.

[16] A. SenGupta and F. Walls, "Effect of aliasing on spurs and PM noise in frequency dividers," *IEEE Proc. Of Frequency Control Symposium*, June 2000, pp. 541-548.

[17] A. M. Fahim, "Jitter Analysis of Digital Frequency Dividers in Communication Systems," *IEEE International Ultrasonics, Ferroelectrics and Frequency Control Conference*, 2004.

[18] J. Crawford, *Frequency Synthesizer Design Handbook*, Artech House: Norwood, MA, 1994

[19] B. Miller and R. Conley, "A Multiple Modulator Fractional Divider," *IEEE Trans. Instrum. Meas.*, vol. 40, pp. 578-583, June 1991.

[20] M. Perrott, T. Tewksbury, and C. Sodini, "A 27-mW CMOS fractional-N synthesizer using digital compensation for 2.5-Mb/s GFSK modulation," *IEEE J. of Solid-State Circuits*, vol. 32, no. 12, pp. 2048-2060, Dec. 1997.

[21] W. Rhee, A. Ali, and B. Song, "A 1.1GHz CMOS Fractional-N Frequency Synthesizer with a 3b 3rd-order ΣΔ Modulator," *Int'l Solid-State Circuits Conference*, pp. 198-199, 2000.

[22] T. Riley, M. Copeland, and T. Kwasniewski, "Delta-sigma modulation in fractional-N frequency synthesis," *IEEE J. Solid-State Circuits*, vol. 28, no. 5, pp. 553-559, May 1993.

[23] T. Nakagawa and H. Nosaka, "A Direct Digital Synthesizer with Interpolation Circuits," *IEEE JSSC,* May 1997, pp. 766-770.

[24] A. Yamagishi, et. al., "A Phase-Interpolation Direct Digital Synthesizer with an Adaptive Integrator," *IEEE Transactions on Microwave Theory and Techniques*, June 2000, pp. 905-909.

[25] J. Ingino, Kaenel, "A 4-GHz Clock System for a High-Performance SoC Design," *IEEE JSSC*, Nov 2001, pp. 1693-1698.

[26] M. Ercegovac, T. Lang, *Digital Arithmetic*, 2004.

[27] H. Nosaka et. al, "A low-power direct digital synthesizer using self-adjusting phase-interpolation technique," *IEEE JSSC*, Aug. 2001, pp. 1281-1285.

[28] D. Johns and K. Martin, *Analog Integrated Circuit Design*, 1997

[29] Tien-Yu, W. et. al., "A Low Glitch 10-bit 75MHz CMOS Video D/A Converter," *IEEE JSSC*, vol. Jan. 1995, pp. 68-72.

[30] A. M. Fahim "A Low-Power Clock Generator for System-on-a-Chip (SOC) Processors," *European Solid-State Circuits Conference*, 2004.

Chapter 8

DESIGN FOR TESTABILITY IN PLLS

8.1 Introduction

Design for testability (DFT) has become an integral design segment of modern digital integrated circuit designs. Analog integrated designs, however, have always lagged behind their digital counterparts in DFT structures. In SoC designs, it is customary to integrate both high performance analog and digital integrated circuits. For this reason, there has been a recent surge in demand for design for testability (DFT) for analog cores in SoC processors [7-10]. In this chapter, DFT issues in SoC PLLs are detailed. First, a characterization of the tests required for SoC PLLs is given. This is followed by a discussion of jitter measurement techniques. This includes a discussion of high resolution off-chip measurement equipment and load board designs. Then, built-in self-test (BIST) techniques for PLLs are given. This includes an explanation of area-efficient integrated test modules and test methodologies that can be adopted to reduce test overhead for PLLs.

8.2 Verification of SoC PLLs

8.2.1 Overview

A typical microprocessor in the past may have included a single PLL that acts a clock generator for all sub-systems on chip. Modern SoC processors, however, have a very heterogeneous set of cores integrated on-chip. Each

core may have differing PLL clock requirements, leading to multiple clock domains on the same die. Moreover, some SoC designs may be quite large and may make clock routing prohibitively expensive and difficult to integrate on-chip. For this class of applications, multiple PLLs may be required.

Multiple PLLs on-chip, however, come with their own challenges. Firstly, integrated multiple PLLs on a chip can lead to coupling of the signal from one PLL onto another resulting in degradation in performance. This, however, can be resolved by judicious placement of PLLs and proper power routing and distribution. Another source of difficulty is that PLLs often consume a large portion of the total chip area. Integrating an excessive number of PLLs on chip can adversely affect the cost of the chip since area penalty translates to increased production costs.

Another challenge for integrating multiple PLLs on chip is the increased test time required for the clock generators. Typically, a single PLL may require only a few milliseconds of test time (which is relatively small compared to other analog integrated circuit cores). If multiple PLLs or other clock generator structures are integrated on-chip, the required test time for the overall clock generation block may be large.

8.2.2 Required PLL Tests

An enumeration of the tests typically required for SoC PLLs is given in this section. Such tests include frequency test, lock time, jitter under different PLL operating modes and frequencies.

Frequency Test. The most fundamental functional test for a PLL is to verify that the output frequency is correct. This is usually done for all values of N (feedback divider ratio) that are used in the PLL. In order to test the frequency of the PLL accurately, the PLL must first be loaded with the correct register settings, then the test equipment must be idle for a certain amount of time to allow the PLL to lock. This amount of time is usually specified by the PLL designer ahead of time. Alternatively, the PLL may be equipped with a lock detection circuitry that can notify the test equipment when it is safe to begin collecting PLL edge samples.

Once the PLL is locked, the edge samples are collected by the test circuitry, which is usually a frequency counter. In its simplest form, a frequency counter is nothing but an accumulator that increments by one whenever an edge (either positive or negative) is generated by PLL. The number of collected samples per fixed time interval, T_{test}, gives an indication of the output frequency of the PLL. After the test time interval, T_{test}, the counter is reset. More specifically, if the number of edges counted is N_{edge}, then the PLL output frequency is given as

$$f_{PLL} = \frac{N_{edge}}{T_{test}} + \Delta f_{err} \tag{8.1}$$

where Δf_{err} is the error in the frequency estimation due to the resolution of frequency measurement accuracy. The accuracy of the frequency depends on the test interval, T_{test}. More specifically, the maximum frequency estimate error can be given as

$$|f_{err}|_{max} = \frac{1}{T_{test}} \tag{8.2}$$

This equation shows that an increase in frequency accuracy entails longer test time. For example, if frequency accuracy of 1kHz is required, this would entail a test time of 1ms plus vector setup and PLL lock time. This test would have to be done for every output frequency of interest, leading to an excessively long test time. In practical frequency tests, the PLL is often first divided down on-chip and then buffered onto one of the chip's digital test pin and the divided down period is measured off-chip.

Lock Time. One important parameter in PLLs for SoC processors is lock time. Lock time is defined as the time it takes for the PLL to produce the desired output frequency to within a certain frequency accuracy. Frequency accuracy is expressed either in Hertz (Hz) or in parts per million (ppm). For example, if the required output frequency is 500MHz and the required frequency accuracy is 1kHz, then a lock time specification can read as 50μsec for 1kHz resolution, or 50μsec for 2ppm resolution.

Measuring lock time, however, can be difficult. One method is to measure the instantaneous frequency of the PLL output and compare it to a high-resolution reference signal tuned at the desired PLL output frequency. Once the frequency has reached the desired resolution, the PLL is considered to be locked. This can be done by external equipment or by using a lock detect circuit.

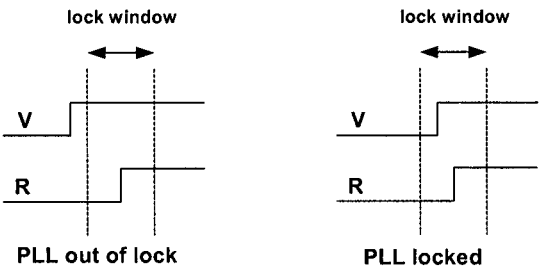


Figure 8-1. Concept of lock detect circuit for PLLs

Most lock detect circuits work by observing the input of the phase-frequency detector (PFD). One of the inputs either the reference, R, or the feedback, V, signal is set as a reference signal and a small Δt timing window is created around this window. If the edge from the other signal (not the reference) falls within this window for M consecutive cycles, then the PLL is considered to be locked. The value of M is usually determined empirically. Figure 8-1 illustrates the concept of lock detection.

The implementation of a lock detect circuit is shown in Figure 8-2 [14]. When the UP and DN output signals of the PFD become high, the difference between the two signals at a time ΔT earlier is captured by the XOR gate. The two signals are shortly reset by the feedback AND gate as explained earlier in chapter 2. If the UP and DN signals are different for a time greater than ΔT, then it is captured by the shift register and a '1' appears at the top most input of multi-input NOR gate; otherwise, the pulse output of the XOR gate is missed by the shift register and a '0' appears at the top most input of the NOR gate. The lock detect output remains low until the phase error is less than ΔT for n cycles, where n is the number of bits in the shift register.

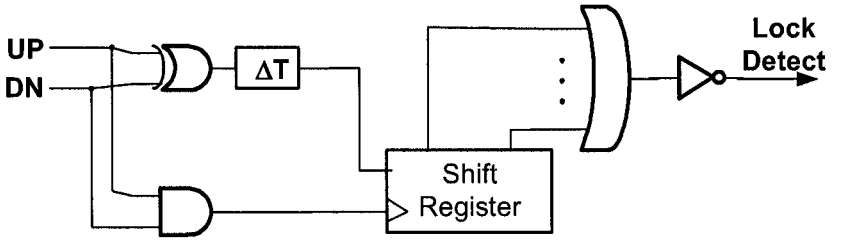


Figure 8-2. Schematic diagram of a lock detect circuit

Jitter Test. Low-jitter PLLs is one of the most quintessential analog blocks in a SoC processor. Jitter affects the timing margin available to the digital processor core as well as the signal-to-noise ratio (SNR) in data converters and other analog clock sampled cores. Jitter must also be verified under different conditions. *Quiet condition* is defined as when there is minimal processor activity and the power supplies are relatively noiseless. Most of the jitter measured under this condition is due to intrinsic PLL jitter or input clock reference jitter. *Noisy condition* is defined as when various parts of the processor are active during the measurement of jitter in the PLL. This would normally involve microprocessor activity, memory accesses, and I/O activity. Depending on the pin assignment on the chip, the I/O activity can result in the most increase in measured jitter from the PLL. This is due to parasitic coupling of large drive I/O signals on to the PLL output. Coupling can occur on the board traces, package, or power supply for the chip's padrings. Figure 8-3 illustrates the possible noise coupling sources.

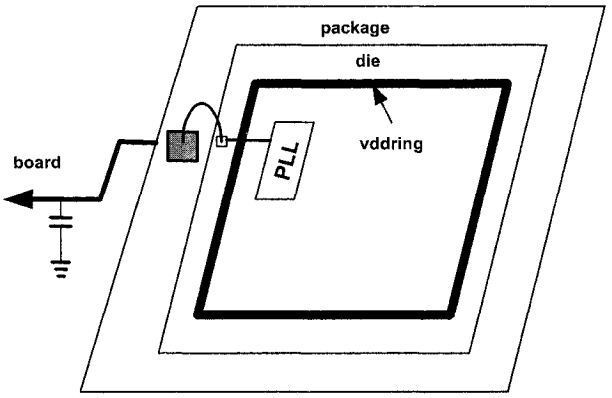


Figure 8-3. Possible noise coupling into the PLL's power supplies

Other conditions are referred to as *functional conditions*. The primary functional condition is the set of desired output frequencies. As the PLL output frequency is varied, the output jitter may vary. Certain PLL architectures can limit this variation, as we've seen in chapter 5. However, they cannot completely eliminate it. This is due to the fact that the noise contribution of other blocks on the chip can change with PLL frequency. Also, there may be clock traces that are too thin that degrade jitter performance at higher frequencies. Furthermore, there may be sections of the chip that are used only when certain PLL output frequencies are generated, resulting in frequency dependent noise.

Other functional conditions include PLL mode of operation. An example of a SoC processor requiring PLL operating mode may be a SoC processor for wireless communications. If voice only mode is required for a certain period of time, the PLL may be put in low-power mode, where the current burned in the VCO is reduced. If multimedia mode of operation is required, the VCO current may be increased to its maximum setting for lowest jitter performance. Other operational modes may include required flexibility in loop bandwidth.

8.3 Jitter Measurement Techniques

Verifying clock jitter performance in PLLs is a major concern in SoC processors. This task is not a trivial issue due to several factors. Modern SoC processors can have clock frequencies measuring from several hundred megahertz well into the gigahertz range. Measuring this signal off-chip can incur a large overhead in terms of on-chip buffering as well as the need for

extra off-chip test equipment and devices. Simply driving the clock signal onto a pad and measuring off-chip can cause the PLL output to couple back into its supplies, through either the board or padring, and hence degrade its own performance. One simple way around this problem is to reduce the output frequency of the PLL through frequency dividers before it is taken off-chip for measurement. The measured signal, however, will include the jitter of the frequency dividers in addition to the jitter of the PLLs.

In this section, high accuracy jitter measurement techniques are reviewed. Off-chip measurements usually involve automated test equipment (ATE) jitter measurement devices. Issues with using ATE measurement techniques are detailed. This is followed by the analysis of the two high-resolution jitter measurement techniques: TIA and digital oscilloscope. The advantages, disadvantages and features of both techniques are detailed.

8.3.1 High-Bandwidth ATE Jitter Measurement Setup

The main objective of jitter measurement of PLLs for SoC is to accurately translate the zero time crossings of the PLL output to the ATE measurement device's pin. This involves low-jitter high-bandwidth load board designs that interface the chip, or device, under test (DUT) and the ATE equipment. There are several issues involved in the load board design which include crosstalk, ground noise, impedance matching, high bandwidth required for to faithfully reproduce the rise/fall times of the DUT output.

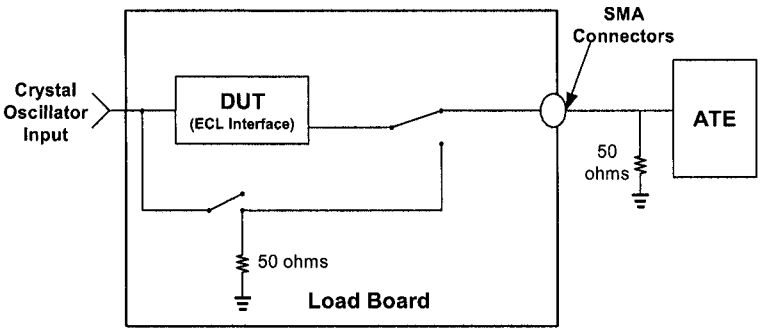


Figure 8-4. Typical load board setup

A typical high-speed load board setup is shown in Figure 8-4 [11]. The output of the DUT assumed to have ECL interfaces. ECL interfaces have the advantages of having well-defined levels and load terminations of 50 ohms and low-swing for fast switching speed. Shown in the setup is the ability to bypass the DUT with its input connected directly with the ATE equipment. This gives the advantage of being able to measure the jitter of a

well established reference signal to accurately calibrate the test equipment. The quality 50 ohm SMA connectors are shown to connect the load board with the ATE. Proper choice of ground connections and avoiding ground loops is important to minimizing the effect of ground noise. Ground noise and bounces can be as high as hundreds of millivolts if proper ground routing is not taken into consideration.

In modern SoCs, however, the output levels coming off-chip are usually CMOS levels. CMOS output logic is problematic in high-speed load board designs, since its required output termination is much larger than ECL termination. This is problematic since any variation in the impedance of the trace between the load board and the ATE equipment can translate to large variations in jitter and timing measurements in general. This would give inconsistent and unrepeatable jitter measurements. One solution to this problem is to provide most of the termination of the CMOS signal on the load board and the remaining 50 ohms to be provided by the ATE. Another solution is to insert a CMOS-to-ECL buffer on the load board to isolate and well-define the termination required for the CMOS signal. This configuration is shown in Figure 8-5. Another advantage of this configuration is the reduced output power level required by the CMOS chip, and hence reduce the effect of coupling of the output signal to the PLL's power and ground supplies.

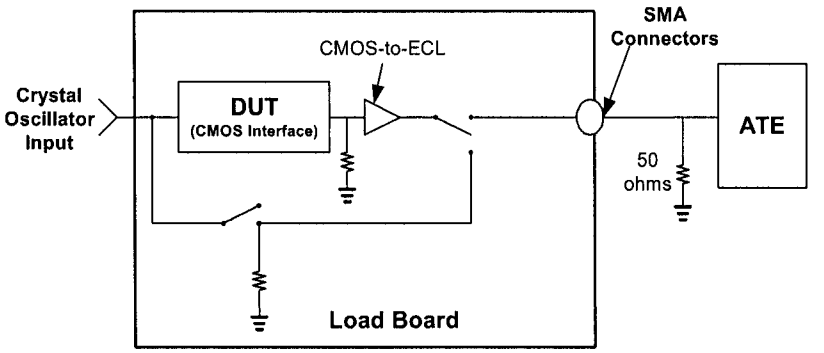


Figure 8-5. Load board setup suitable for DUTs with CMOS output signal levels

High-bandwidth is also a necessary feature of the load board interfacing the DUT to the ATE. High-bandwidth can be achieved, in part, by proper termination. To help enhance termination, resistors with 1% tolerance must be used; also resistors with wattage ratings of at least five times maximum expected output power levels help reduce the effects of thermal drift. Proper termination avoids ringing, which in some cases may alter the zero crossings of the signal. The use of high-quality SMA connectors also helps increase the bandwidth of the measurement system. Another technique that helps to

increase the bandwidth is to use an attenuator to reduce the amplitude of the signal from the DUT. In doing so, however, care must be taken to correctly calibrate the DC threshold level of the ATE jitter measurement equipment. One must also ensure that the ATE jitter measurement device has sufficient DC level resolution to accurately capture the zero crossings for low-voltage swing measurements [12].

8.3.2 ATE Jitter Test Equipment

In this section, different techniques used by various test equipment to measure jitter are reviewed. PLL jitter test equipment can vary in sophistication ranging from an analog oscilloscope to a digital oscilloscope equipped with high data rate sampling and powerful digital signal processing algorithms. The choice of proper test equipment will depend on the required jitter measurement accuracy, which will increase with increasing PLL output frequencies.

Analog Oscilloscope. An analog oscilloscope is the most basic instrument than can be used for jitter measurement. The PLL output is displayed on the oscilloscope and triggered with itself. Once the PLL output is triggered, a range of output periods is displayed on the screen in a continuous fashion and the waveform is wrapped around itself continuously. This means that in the case of zero jitter, the zero crossing appearing on the screen would be at a fixed location. With jitter, however, the zero crossings on the screen would appear to fluctuate randomly with time. The persistence setting on the oscilloscope is then set to "infinity." This setting causes each output signal displayed on the screen to remain there and not be cleared with the next range of input data. Over time, a waveform, as shown in Figure 8-6, is formed on the display screen. The maximum peak-to-peak jitter is then readily measured by measuring the thickness of the signal at zero crossing. Time interval error (TIE), or absolute jitter, can then assumed to be half the maximum peak-to-peak jitter.

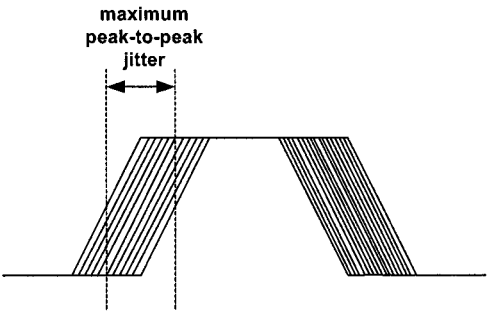


Figure 8-6. Typical clock signal measured oscilloscope with infinite persistence

There are several problems with this method of jitter measurement. First, the number of samples collected is not well controlled. As a result, the jitter measured using this procedure is usually unrepeatable. Furthermore, jitter measurement relies on "eye balling" the jitter on the oscilloscope. This method cannot be used for production testing. Also, the assumption that TIE is half the maximum peak-to-peak jitter is not always true. It is only true for a Gaussian distribution with zero mean. In general, deterministic jitter may be present in the PLL output signal, and thus creating an asymmetric jitter probability distribution.

Time Interval Analyzer. Another popular method to analyze jitter is to use a time interval analyzer (TIA). TIA measurement very much resembles how we measure time using a stopwatch. There is a start time, stop time, and the time interval between the start and stop times is measured relative to an internal high frequency sampling signal. Figure 8-7 illustrates how a TIA measures time [2]. As the figure shows, a high-frequency high-quality reference sampling clock signal is required.

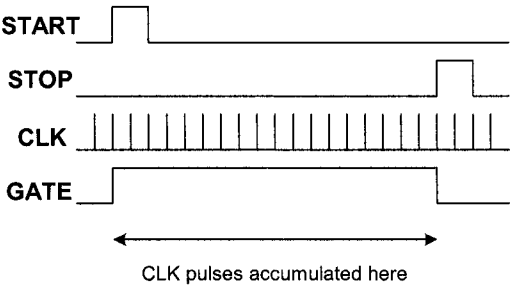


Figure 8-7. Illustration of TIA jitter measurement method

There are certain terms that are defined for TIA measurements. *One shot* measurement is defined to be the time interval between start and stop pulses. Higher sampling clock leads to higher measurement resolution. For example, if the sampling frequency is 1GHz, accuracy is limited to 1ns resolution. *Minimum interval* is defined as the minimum time between start and stop pulses which the counter will recognize. The minimum *dead time* is the time from the stop pulse to the acceptance of the next start pulse. The *minimum pulse width* is the shortest pulse the counter will recognize as a start or stop pulse. The bandwidth of the amplifiers connecting the start/stop pulse generators and the pulse counter determines the minimum pulse width.

Limitations of TIA come from the maximum internal clock sampling frequency. This can be overcome by one of two techniques. The first technique is known as time interval averaging. In this method, several time interval samples are taken and averaged. The resulting resolution is

improved by $\sqrt{N}$, where N is the number of time interval samples [2]. Although useful for increasing accuracy of measuring a PLL output period, it destroys any phase jitter information.

Another technique to increase accuracy in TIAs is to use digital interpolation. Digital interpolation makes use of a device called a *phase-startable and phase-lockable oscillator* (PSPLO). This device is capable of producing a frequency that is incrementally smaller than the sampling frequency and is phase locked with the sampling frequency at a certain time instant. The frequency-shifted signal can be produced instantaneously and locked in phase with the sampling clock signal. Start and stop pulses with periods slightly larger than the sampling clock are then synthesized using the PSPLO as shown in Figure 8-8. The measurement interval is from the end of the start pulse to the end of the stop pulse. The entire time interval is then given as

$$TI = T_1 - T_2 + T_3 = T_0 \cdot (N_0 + (1 + 1/N) \cdot (N_1 - N_2)) \tag{8.3}$$

where N_1 is the number of start oscillator pulses, N_2 is the number of stop oscillator pulses, and N_0 is the number of sampling clock pulses between end of start and end of stop pulses.

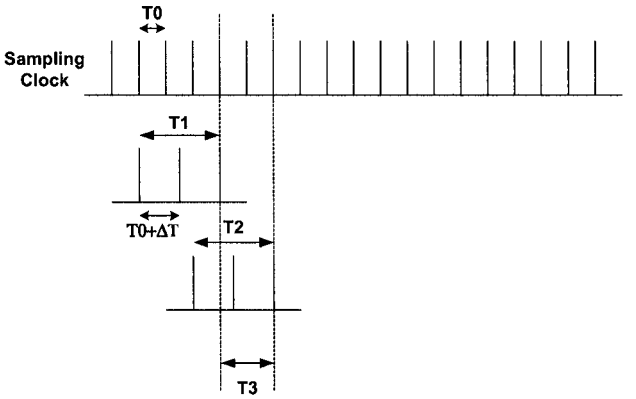


Figure 8-8. TIA measurement device using digital interpolation

Another limitation in TIA measurement techniques comes from the fact that frequency information is destroyed in the measurement. This is because jitter is not measured on consecutive cycles basis. Raher the one shot interval may span several PLL clock periods. Care must be taken when choosing the one shot timing interval. If the one shot is a multiple of the PLL period being measured, or a period of a periodic jitter signal present in the PLL output, then this may lead to incorrect jitter measurement. The jitter

sample space must be large enough and the one shot interval must not be harmonically related to any signal present in the PLL output.

Furthermore, if the PLL output contains deterministic jitter (DJ) as well as random jitter (RJ), then the probability distribution of the total jitter (TJ) at the PLL output a convolution of the two different probability functions. This is assuming that the random jitter is independent from the deterministic jitter, which is true in most cases. A sample of a histogram of a PLL output jitter containing Gaussian distributed RJ and a single spur generating DJ is shown in Figure 8-9. As the figure shows, the total output histogram is a convolution of the two independent histograms.

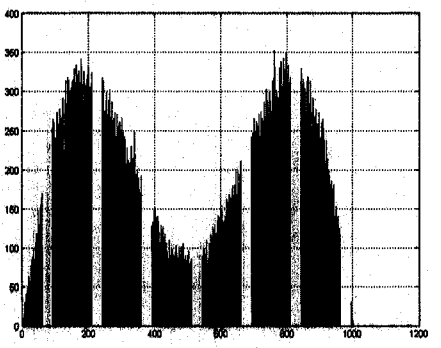


Figure 8-9. PLL output histogram with RJ and DJ

In general, several sources of DJ may exist in the PLL output and clear indication of presence of DJ may not be as obvious. In order to guarantee accurate measurement, a large enough sample must be used to account for all sources of jitter. This means that the required sample space for one RJ source must be multiplied by the number of DJ sources. This can quickly increase the sample space required for TIA measurements.

TIA RJ/DJ Separation. One technique that has recently gained popularity is to separate random jitter and deterministic jitter. One reason for doing so would be to reduce the number of samples required for measurement. Other reasons include the analysis of jitter in PLL output and identifying sources of jitter to help reduce jitter.

Wavecrest has recently introduced a TIA technique for RJ/DJ separation [15]. The underlying assumption in this technique is that RJ and DJ jitter sources are independent and the output TJ distribution is the convolution of all the independent jitter sources. The separation technique uses a proprietary algorithm called "TailFit". The algorithm first finds the tail region of the histogram, then fits the data with a Gaussian distribution that best coincides with the tail region. Left and right regions of the histogram can be tail fitted separately. The random jitter components for each tail fit

are then averaged to represent the random jitter component of the total jitter. The chi-squared statistic is used to determine the quality of the fit. The tail fit is recursively enhanced until the quality of the fit is determined to be satisfactory The number of samples required now is equal to the number of samples required to capture the longest period of deterministic jitter and the number of samples required to get a high degree of confidence to estimate the standard deviation of the Gaussian distribution of the random jitter component. The maximum peak-to-peak jitter can then be estimated to be fourteen times sigma plus deterministic jitter.

Real-time Digital Oscilloscope Testers. Another class of equipments for measuring jitter is real-time digital oscilloscopes. This technique relies on the amplitude sampling of the test waveform by a high frequency sampling signal (may be as high 20Gsamples/sec). In order to meet the Nyquist criterion, an input signal must be sampled at least twice every period. However, if a signal is an over-sampled bandlimited signal, then the sampled signals can be used to mathematically model the original signal using sin(x)/x interpolation [1]. This interpolation enhances the resolution of the sampled signal.

The resolution attainable by real-time digital oscilloscope testers can be somewhat higher than TIA methods. The resolution of digital oscilloscopes is limited by the sampling rate (as high as 20Gs/s), signal interpolation accuracy (using sin(x)/x approximation), and mathematical signal processing manipulation on the sampled waveform. Using these techniques, timing resolution of tens of femtoseconds is possible. In TIAs, the timing resolution is limited by the maximum achievable internal sampling clock generator. Timing resolution in TIAs is currently limited to hundreds of femtoseconds.

Another limitation in accuracy in timing measurement systems is *jitter noise floor*. Jitter noise floor is defined as the jitter measurement with no signal applied. This lower limits practical resolution needed by the jitter measurement system. For example, Tektronix's Tek TDS8000B has a 0.8ps trigger jitter and jitter noise floor [1]. Using interpolation to attain better resolution does not have any advantage.

Random jitter (RJ) and deterministic jitter (DJ) can also be separated using digital oscilloscopes. This has been demonstrated by Tektronix [1]. The technique relies on a proprietary spectral analysis method to perform RJ and DJ. It is possible to perform spectral analysis in this case, since accurate sampling of the input signal has been performed.

8.4 Design for Testability and Self-Test in PLLs

One method of alleviating test concerns for PLLs is to have built-in test structures in the PLL design. Designing the PLL with testability in mind greatly eases the test and debug of PLLs. Figure 8-10 shows a fully observable and controllable PLL. This has great advantages when debugging a PLL design and leads to swift and cost effective methods of identifying flaws in the design and methods of correcting the design.

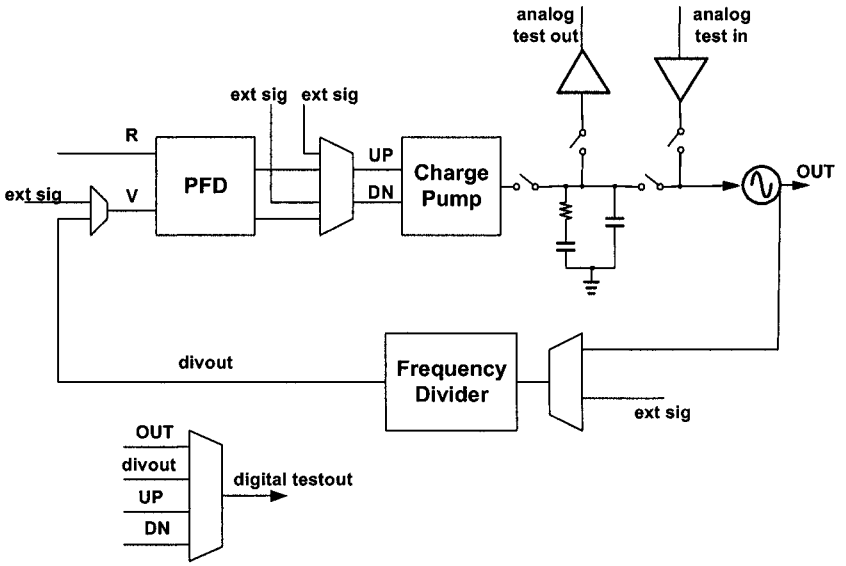


Figure 8-10. Fully observable and controllable PLL design

Although useful for debugging, it has less use for testing. In design for testability (DFT), structures that can automate the verification of functionality and performance of the PLL are needed. In this section, a variety of DFT structures for PLL design are explored.

Apart from aiding in automating the test procedure for PLLs, there are other practical reasons why DFT is important for PLLs. PLLs for SoC processors may be available as hardmarcros offered by various vendors and manufacturers as a hard IP (intellectual property). In many cases, the provider of the PLL hard IP may wish to encrypt the design for IP protection. In this case, the interface as well as method for testing the PLL must be well-defined. In this section, techniques for on-chip functional and performance PLL testing is described.

8.4.1 Functional Test

The simplest PLL test is a frequency accuracy test. This can be easily performed by having an on-chip accumulator and digital comparator. This is similar to the technique described earlier in section 8.2.2, but all circuitry is built on-chip. The circuit then outputs a single-bit verifying whether or not the frequency accuracy test succeeded. Figure 8-11 shows an example of a frequency accuracy tester circuit. If the circuit outputs a '1' for k consecutive reference cycles, where k is an arbitrary number, then the PLL is considered locked.

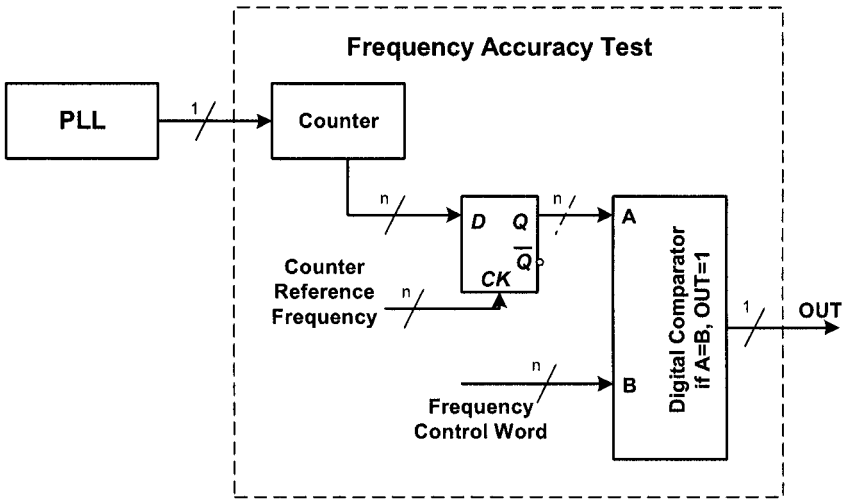


Figure 8-11. Schematic diagram of frequency accuracy test circuit

8.4.2 Indirect Jitter Measurement Techniques

On-chip jitter performance test is more complex and requires precise on-chip phase error detection and quantification. Although several techniques exist that can be integrated on-chip, these techniques tend to be expensive in terms of area and power consumption [8-9,13]. These techniques often rely on power and area hungry methods of generating very accurate phases of a given moderate frequency clock in order to sample an incoming signal with very high timing precision.

One method of evaluating the performance of a PLL is to measure its jitter transfer function [6,7]. This is done by first inserting jitter into the PLL then verifying that it does not exceed a certain maximum jitter limitation.

In order to measure the jitter transfer function, the input to the PLL must be sufficiently jittery. Jitter generation can be performed by the circuit shown in Figure 8-12. The input phase is dithered by a digital sigma-delta

modulator. The low-pass characteristic of the PLL filters the quantization noise.

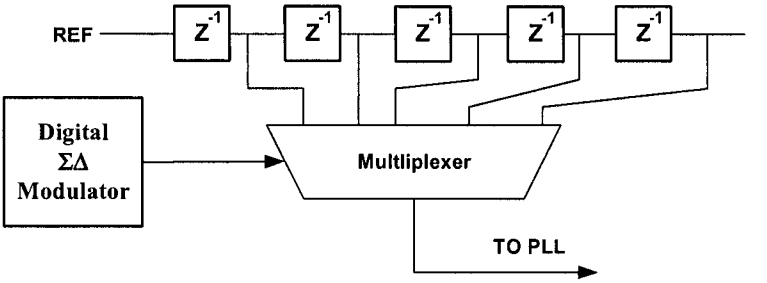


Figure 8-12. On-chip jitter generation circuit

The second step is to quantify the amount of resulting output jitter and compare it to a certain threshold level for jitter. If a test clock that is eight times the frequency of the PLL output clock rate is available, then jitter phase error can be verified to be within $\pm\pi/8$ radians. In general, designing a jitter threshold circuit is not a trivial task.

In some cases, the major source of jitter is the power supply. In this case, an on-chip method of noise injection to the power supply must be provided. Figure 8-13 shows an on-chip power supply noise generator [7]. This circuit is capable of generating both low-frequency noise and high-frequency noise. When high frequency noise is desired, the top branch is active. Noise can be generated by a sigma-delta modulator as shown previously or from a linear feedback shift register (LFSR). At each PLL output edge, a pulse is applied to the NFET switch and hence the PLL turn supply is disturbed. The magnitude of the power supply ripple depends on the duration of the pulse.

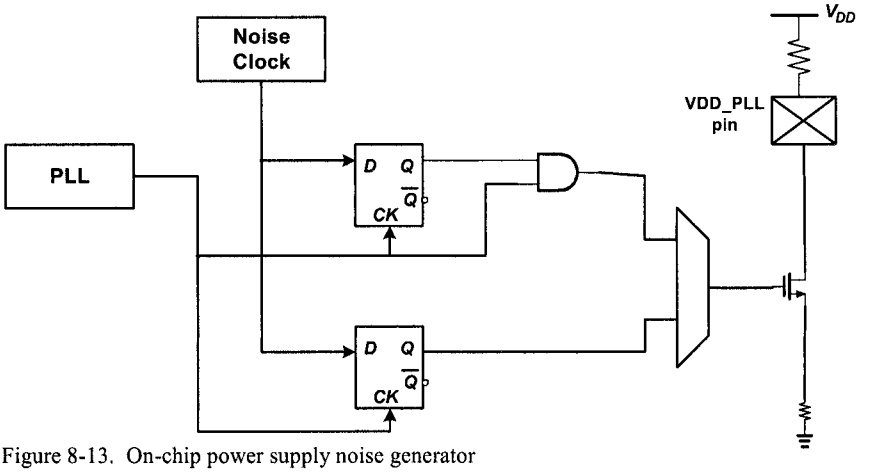


Figure 8-13. On-chip power supply noise generator

If lower frequency noise is required, the lower branch is activated. This creates voltage steps on the supply as opposed to small noise ripples. This is useful in evaluating the power supply sensitivity to large spikes introduced on the supply when other circuits power on or off. After injecting this noise source, a jitter threshold circuit similar to the one described earlier may also be used to evaluate the PLL performance.

Yet another technique relies on a fault simulation test, as opposed to a functional or performance test [3]. The open loop response of the PLL is evaluated and verified to lie within an upper and lower threshold set by the allowed process variation in the process. If excessive process variation occurs, then this test will indicate a FAIL state and the die will be considered defective. The reason why the open loop response is measured is that most process variation occur from devices in the open loop path of the PLL (loop filter, VCO, charge pump).

In this test, input test vectors are inserted into the PFD. The feedback counter measures the resulting VCO period. In the case of the loop filter voltage ramping up or down, the feedback divider will measure the average frequency of the VCO over the test period range. This assumes that the VCO gain is linear over the applied loop filter voltage range. The input vectors of the PFD are set such that the loop filter is first discharged then charged. Figure 8-14 shows a typical curve evaluating the open loop response of the PLL. As long as the count value lies between the upper and lower bounding curves, the PLL is considered to be fault-free.

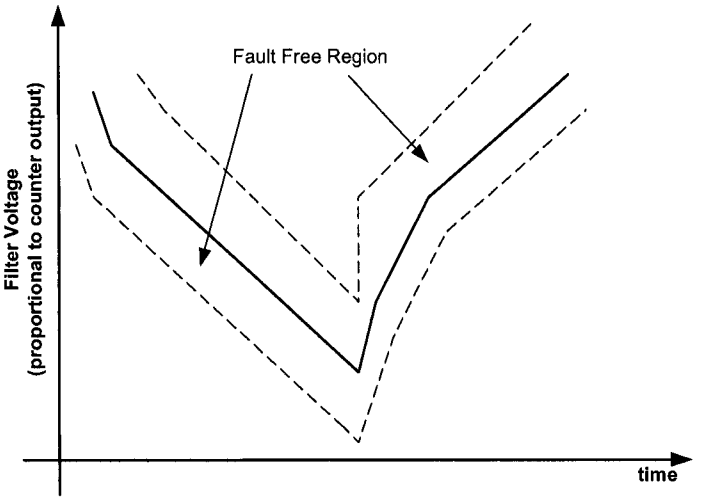


Figure 8-14. Output test response of fault simulation test technique

8.4.3 Precise On-Chip Jitter Measurement

One limitation with the above techniques is that they rely on indirect jitter measurement technique. For instance, the jitter transfer function of the PLL or its power supply sensitivity is measured. The disadvantage of these techniques is that an artificial source of jitter is applied, either at the input or through the supplies, then it is assumed to be the dominant source of jitter and jitter is evaluated. In practice, however, it is rarely the case that there is only one source of jitter in a PLL. There may be a substantial contribution of jitter from both the power supplies as well as the input reference signal. Furthermore, the amplitude of the applied noise may not be realistic. In fact, an excessive amount of noise may be required in order to ease the implementation of the jitter threshold test circuit.

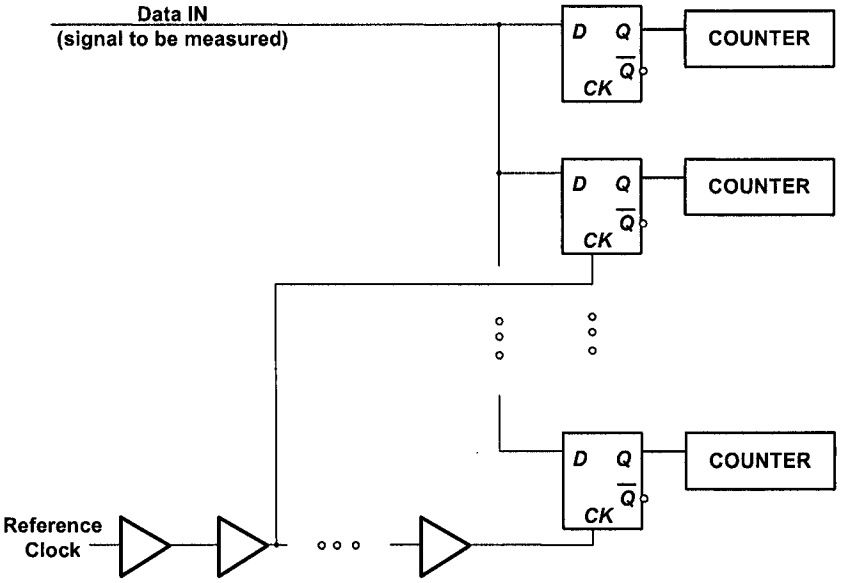


Figure 8-15. Basic on-chip jitter measurement circuitry

An alternative method of testing jitter on-chip is to attempt to directly measure jitter [4]. One method of accomplishing this task is to use a delay line, set of D-type flip-flops (DFFs), and a bank of counters, in the configuration shown in Figure 8-15. The input clock signal is a high-resolution reference clock, the data is the measured signal (output of the PLL under test). The clock signal is intentionally delayed with respect to the PLL signal. The delay line is chosen to be long enough to capture the full range of required positive and negative phase errors. For example, if it is anticipated that the phase error will range from –1ns to +1ns and the delay in each cell of the delay line is 50ps, then at least 40 delay cells are needed in

the delay line. As the clock signal is delayed further, the probability of a '1' being latched into the corresponding DFF becomes higher. The resulting data in the counters will correspond to the cumulative probability density function (CDF) of the jitter distribution.

The two disadvantages of this technique are large area and limited resolution. Large area is incurred by the fact that a large number of counters and the lengthy delay line. Furthermore, the resolution of the delay line varies with process and operating condition and is limited to the delay of on delay cell stage. These disadvantages can be overcome by using a Vernier delay line (VDL), shown in Figure 8-16, and a more efficient implementation of the jitter measurement setup, shown in Figure 8-17.

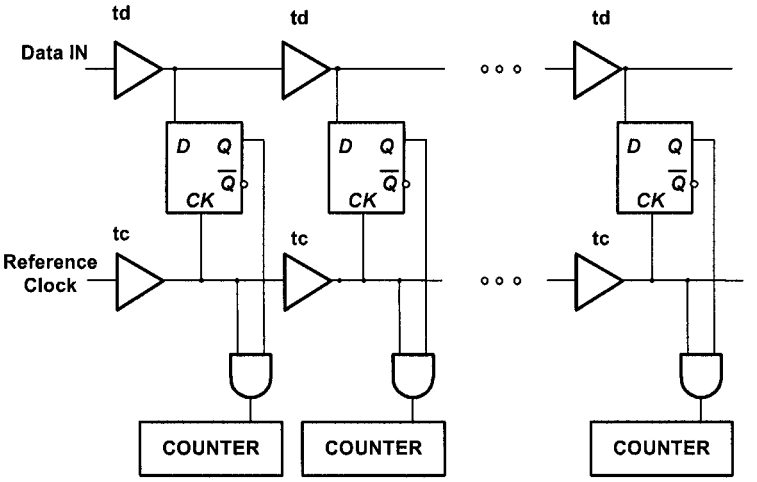


Figure 8-16. Vernier Delay Line

A Vernier delay line produces a delay that is proportional to the difference in delay of two closely size buffers. As shown in Figure 8-16, the delay difference in each stage grows by an amount proportion to t_d-t_c. The disadvantage with this technique is that more delay stages are needed to cover the same range as before. Using the previous example, if t_d-t_c is 5ps, then 400 times 2, or total of 800 delay stages, as opposed to 40 delay stages, are needed to implement the VDL. This consumes a large amount of area and mismatch between the delay cells becomes difficult to control.

These difficulties can be overcome by using an efficient single pair of inverter stages in the VDL as shown in Figure 8-17. Furthermore, the accumulators are all merged into one accumulator that now holds a value proportional to the jitter during one reference cycle. The inverters connected in ring oscillator configuration do not accumulate any appreciable jitter since the accumulation loop is broken for a period equal to half the

reference period (assuming the input waveform is a 50% duty cycle waveform).

Using this technique, an rms noise floor of less than 90ps has been reported [4]. This is in comparison with less than 25ps noise floor achievable with a Wavecrest DTS-2770 jitter analyzer. Although the Wavecrest jitter analyzer is capable of 0.8ps jitter noise floor, the extra jitter was most likely introduced by the test setup. The reported area of the on-chip jitter measurement circuit was only $0.12mm^2$.

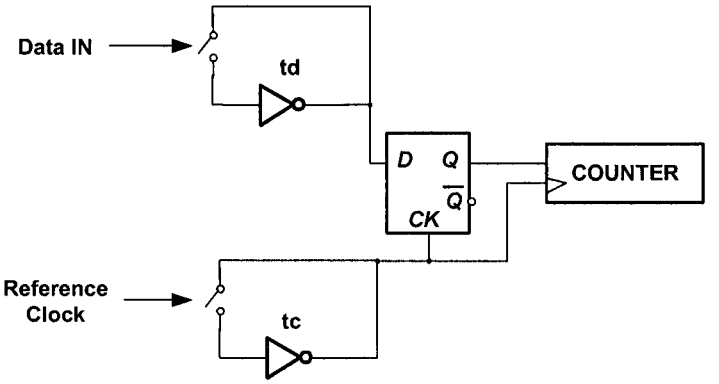


Figure 8-17. Efficient on-chip jitter measurement circuit using VDL

8.5 Summary

In this section, design for testability in PLLs has been covered. The first step in testing a PLL is defining the required tests. The required tests are frequency accuracy test, lock time test, and jitter test. Next, off-chip jitter measurement techniques including load board setup suitable for low-jitter CMOS clock signals have been reviewed. Some insights into how modern jitter analyzers measure jitter has been given. This includes time interval analysis (TIA) methods and digital oscilloscope sampling. Techniques to enhance resolution and speedup jitter measurement times such as interpolation and random jitter / deterministic jitter splitting techniques have been reviewed for both TIA methods as well as digital oscilloscope methods.

Design-for-testability (DFT) for PLLs, and for analog IP cores in general, is becoming more important for several reasons. Firstly, testing these cores as processing frequencies sore becomes more difficult. Secondly, as more analog IP cores become commercialized and are encrypted, the need for a "visible" wrapper around the analog IP core is apparent. This includes a well-defined interface for testing the analog IP core. In order to limit test time, a wide variety of on-chip jitter testing approaches has been given. This

ranges to simple indirect methods to more elaborate direct methods employing Vernier delay lines.

The choice of the amount of DFT to be added to the PLL core must be decided based on several factors. First, it must be kept in mind that the test time of a single PLL chip design is only a few milliseconds. This is a very small amount of time, and only very limited BIST would be required. However, as the number of PLLs per die increases, or if a large suite of PLL output frequencies must be tested, then more BIST may be required. Also, whether a dedicated pin for PLL testing is available or not can affect the amount of BIST that must be covered on-chip.

REFERENCES

[1] "Tektronix TDSJIT3 Seminar," Application Note, Tektronix, 2003.

[2] "Fundamentals of Time Interval Measurements," Application Note 200-3, Agilent Technologies.

[3] S. Kim and M. Soma, "An All-Digital Built-In Self-Test for High-Speed Phase-Locked Loops," *IEEE Trans. On Circuits and Systems II*, vol. 48, no. 2, pp. 141-150, February 2001.

[4] A. Chan and G. Roberts, "A Jitter Characterization System Using a Component-Invariant Vernier Delay Line," *IEEE Transactions on VLSI*, vol. 12, no. 21, pp. 79-95, January 2004.

[5] T. Egan and S. Mourad, "A Framework for the Characterization and Verification of Embedded Phase-Locked Loops," *IEEE Trans. On Instrumentation and Measurement*, vol. 51, no. 6, pp. 1234-1239, December 2002.

[6] B. Veillette and G. Roberts, "On-Chip Measurement of the Jitter Transfer Function of Charge-Pump Phase-Locked Loops," *Int'l Test Conference*, pp. 776-785, 1997.

[7] K. Jenkins and J. Eckhardt, "Measuring Jitter and Phase Error in Microprocessor Phase-Locked Loops," *IEEE Design & Test of Computers*, pp. 86-93, April-June 2000.

[8] T. Yamaguchi, et. al, "Jitter Measurements of a PowerPC™ Microprocessor Using an Analytic Signal Method," *Int'l Test Conference*, pp. 955-964, 2000.

[9] T. Yamaguchi, et. al., "A Method for Measuring the Cycle-to-Cycle Period Jitter of High-Frequency Clock Signals," *Int'l Test Conference*, pp. 102-110, 2001.

[10] A. Crouch, et. al., "The Testability Features of the 3rd Generation Coldfire® Family of Microprocessors," *Int'l Test Conference*, pp. 913-922, 1999.

[11] D. Petrich, "Achieving ±30ps Accuracy in the ATE Environment," *Int'l Test Conference*, pp. 691-700, 1994.

[12] M. Williams, "A Discussion of Methods for Measuring Low-Amplitude Jitter," *Int'l Test Conference*, pp. 646-652, 1995.

[13] B. Arkin, "Realizing a Production ATE Custom Processor and Timing IC Containing 400 Independent Low-Power and High-Linearity Timing Verniers," *Int'l Solid-State Circuits Conference*, pp. 348-349, 2004.

[14] A. Atriss, B. Beterson, L. Parker, "Enhanced Voltage-Controlled Oscillator and Lock Detect Circuit," *IEEE 35th Midwest Symposium on Circuits and Systems*, vol. 1, pp. 594-596, 1992.

[15] "A New Method for Jitter Decomposition Through Distribution Tail-fitting," Technical Bulletin #9, Wavecrest.

Chapter 9

CLOCK PARTITIONING AND SKEW CONTROL

9.1 Introduction

The clock rate of system-on-a-chip (SoC) processors has been steadily increasing. This has been fueled primarily by rapid reduction in feature length in CMOS technologies as well as the increased availability of high-performance IP blocks. As feature lengths are reduced to the nanometer range, however, new design challenges arise. Gate delays no longer dominate overall delay in digital designs. Moreover, inductive effects in high-frequency lines, such as clock distribution networks, become important as frequencies soar up to the multi-gigahertz regime. In nanometer technologies, the termination of long clock routes is no longer capacitive due to gate leakage effects. An overall understanding of the design issues from a chip-level perspective becomes important.

This chapter gives an overall framework of the design multi-gigahertz clock distribution networks within the context of system-on-a-chip (SoC) processors. First, SoC clock distribution networks and recent trends are discussed. Next, theoretical foundations of clock distribution networks and the fundamental limits on their performance is discussed. This is followed by a survey of recent active skew management strategies. Low-power clock distribution strategies are then reviewed. As SoC processors become more heterogeneous, so does their clock management system. In the last section, multi-clock domain interfacing issues are discussed.

9.2 Clock Distribution Networks in SoCs

The simultaneous reduction of feature length of CMOS technologies and exponential increase in processor speeds has become the norm over the past few decades and is expected to continue so for years to come. Clock distribution networks has become critical in maintaining the performance growth. The two main design parameters are clock skew and clock jitter. Clock skew is defined as the variation of timing signal delay in the clock distribution network [1]. Clock jitter is defined as the variation in clock period with respect to the targeted period. In recent years, clock power has also been an important design parameter to minimize.

Fig. 9-1 shows a typical topology of a clock network in a modern SoC processor. It consists of a centralized clock generator device, typically an on-chip phase-locked loop (PLL) [2]. A two-level clock distribution approach is typically adopted to minimize skew, jitter, and power consumption. In the first level, the PLL's output is distributed globally to well-defined zones or clusters of digital logic. The second level clock network consists of local clock distribution, where the global clock signal is routed to the local flip-flops. The optimization of the two levels often differs. In some instances, for example, intentional skew in a local clock network may actually give more timing margin to a critical path in a synchronous block. Also, in SoCs, different IP blocks may require different frequencies, and hence necessitate the use of frequency dividers or MNA counters between the first and second levels of the clock networks.

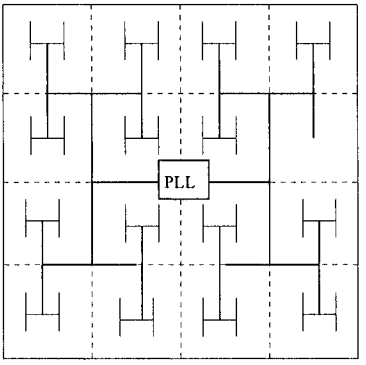


Figure 9-1. Modern clock distribution network.

Fig. 9-2 shows the fraction of clock period consumed by jitter and skew versus operating period for modern processors [14]. The trend line shows a 1/x relationship between period and skew. This shows a rapid increase in percentage of clock period consumed by clock skew and jitter.

Extrapolating this trend to higher frequencies reveals that clock jitter and skew will occupy half of the clock period of processors when operating at 15.6GHz. Theoretical foundations of this trend are explained in the next section.

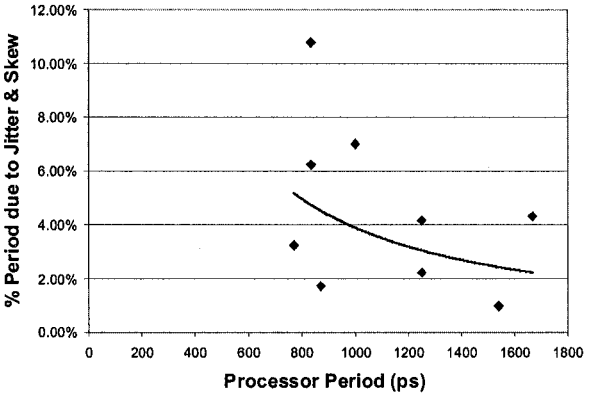


Figure 9-2. Fraction of clock period occupied by jitter and skew

9.3 Performance Limitations in Clock Networks

In order to understand the performance limitations on SoC clock networks in the nanometer range, some theoretical analysis of the underlying technology (interconnect and FET devices) is necessary.

The DC equations for capacitance and resistance in a wire are

$$C_{wire} = \frac{\varepsilon}{t_{ox}} \cdot A_{wire} \tag{9.1}$$

$$R_{wire} = \rho \cdot \frac{L_{wire}}{A_{wire}} \tag{9.2}$$

respectively, where L_{wire} is wire length, A_{wire} is wire area, and t_{ox} is dielectric thickness between a wire and another conductor. Using QCV scaling rules [3], one can easily determine that RC delay in a wire does not change with technology. This represents a great challenge as technology scales since this means that for gigahertz processors, clock latency can easily be several clock periods long. SoC processors typically use heterogenous clock domains with different operating frequencies making synchronization between different IPs within a SoC even more challenging.

Another challenge is the fact that at gigahertz frequencies, clock delay is no longer determined by a simple RC delay; wire inductance starts to become a significant component in the overall clock routing impedance. Other high important frequency effects include skin effect, line proximity

effect, and loop proximity effect [4]. Skin effect is the crowding of current near the edge of the conductor when no other conductors are present. This results in a reduction in the effective thickness of a wire. Modern digital processes are usually well optimized to minimize this effect. Line proximity effect is the crowding of current in a conductor on the side facing another conductor. This effect is more important in nanometer technologies as the metal pitch becomes larger than its width as shown in Fig. 9-3. It also results in an induction of current flowing in the reverse direction in an adjacent conductor. The line proximity effect results in a frequency dependent wire resistance given as [4]

$$R_{eff}(f) = R_{DC} \cdot \left(1 + \frac{1}{2} \left(\frac{\mu_0 \cdot W^2}{R_{sheet} \cdot P} f \right)^2 \right) \tag{9.3}$$

where R_{DC} is the DC resistance of the wire (no inductance effects), μ_0 is the permeability of free space, P is the wire pitch, R_{sheet} is the sheet resistance and W is the wire width. As equation (9.3) shows, wire resistance increases with frequency. Loop proximity effect refers to the frequency dependence of the mutual inductance of a wire on nearby wire lines. As frequency increases, the mutual inductance to closer wires increases, whereas the mutual inductance to farther wires reduces.

The key to minimizing skew due to inductance is to accurately control the inductance return paths in clock distribution networks. To this end, some design guidelines are necessary. First, wide lines are usually split into multiple fingers interspersed with power supply and ground shields. This is to provide a well defined return path to the clock lines. Clock lines are also usually aligned with the power grid or confined to within a power grid. This usually makes it simpler to analyze the inductance return path of a clock line. Another method is to route a clock signal differentially. The advantage of this technique is that line proximity effect actually helps to boost the clock signal. The disadvantage is extra power dissipation and area consumed by differential clock routing.

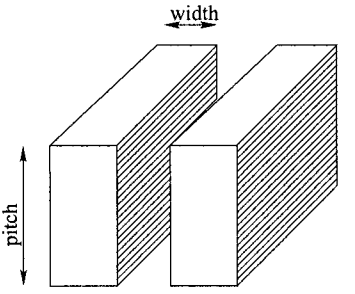


Figure 9-3. Cross section of 2 metals in a nanometer technology

Most analyses of clock distribution networks ignore ringing effects and their impact on device reliability. In general, a wire in a nanometer technology can be modeled as shown in Fig. 9-4. R_s, L_s, and C_s represent the series resistance, series inductance, and capacitance to silicon interface, respectively. R_{sub} and C_{sub} are the parasitic resistances and capacitance from silicon interface to the ground pins. These parasitics can usually be attenuated by inserting sufficient substrate connections. R_{drive} is the source load, which is typically the ON resistance of a CMOS inverter. C_{LOAD} is the load capacitance, which is typically the gate capacitance of several CMOS inverters.

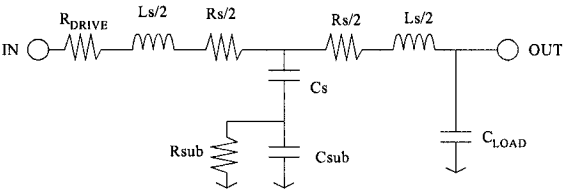


Figure 9-4. Model of a clock wire strip in nanometer technology

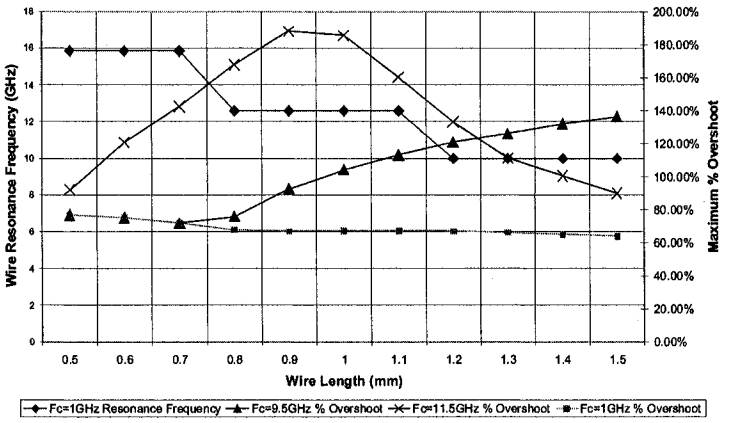


Figure 9-5. Variation of voltage overshoot in a clock line

Fig. 9-5 shows the effect of transmitting a 1V clock signal through a wire of varying length in a 90nm technology at various frequencies. R_{drive} was assumed to be 16Ω and C_L is 0.25pF in these simulations and 1GHz, 9.5GHz, and 11.5GHz clock frequencies were used.. As the figure clearly shows, the percent overshoot can be quite large. The amount of overshoot generally decreases with frequency due to increased resistive loading. However, when the operating frequency is close to the wire resonance frequency, the amount of overshoot increases quite dramatically. This can

have serious consequences to gate-field breakdown. The overshoot can be reduced by increasing the load capacitance or the clock driver's source resistance.

Another important effect in nanometer CMOS technologies is gate leakage [15]. Gate leakage can be modeled as a parallel resistor to the gate input of a FET device. Simulations show that this resistance is only important when it is comparable to wire resistance, which is quite small. Another effect of gate leakage is that it adds a pole at $R_{LOAD}C_{LOAD}$, where R_{LOAD} is the resistance due to gate leakage. If this pole is close to the self resonance frequency of the wire, its effect can be significant. However, since R_{LOAD} and C_{LOAD} are inversely proportional, the pole introduced is usually far removed from the self resonance frequency of the wire. These results are summarized in Fig. 9-6. The transient operation of the clock signal is only affected when the gate leakage is larger than 1mA.

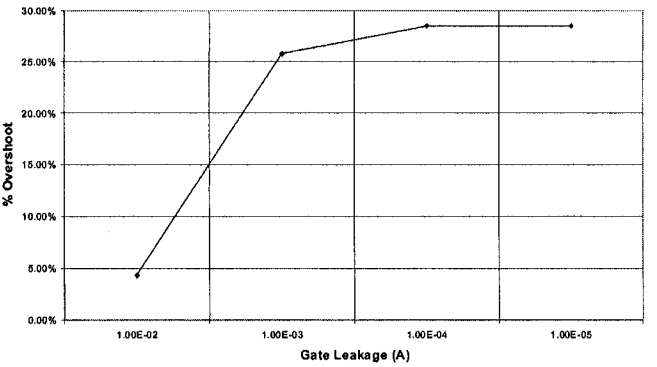


Figure 9-6. Effect of Gate Leakage on clock transmission line

The minimum skew obtainable is limited by the speed of light in silicon technologies, which is given roughly as 150μm/ps [5]. The signal in a local clock network is limited to the maximum skew budget. Furthermore, to avoid a low frequency pole introduced by the parasitic capacitive load in the local clock network, the product of Z_0C_{LOAD} should be minimized, where Z_0 is the impedance of the local clock network. Therefore, the constraint here is

$$l_{local,\max} = 150\mu m / ps \cdot [\max skew - slack] \tag{9.4}$$

The global clock network, on the other hand, suffers from jitter and skew caused by PVT variations in the buffers inserted into the network. In order to minimize jitter and skew, it is desirable to make global clock lines are long as possible. Considering Z_0 being the parasitic inductance and capacitance shown in Fig. 9-4, it can be shown that to send a full-swing signal, the maximum length of a wire is given as

$$l_{global,max} = 2\left(\frac{Z_0}{R_{drive}}\right)\cdot \ln\left(\frac{4Z_0}{Z_0 + R_{drive}}\right) \tag{9.5}$$

This shows that the maximum wire length is increased by increasing Z_0 while minimizing R_{drive}. This can be achieved by using wider lines for clock transmission wires. Using (9.4) and (9.5) and the total length of the clock wirelength, the minimum number of buffers can be derived. The worst case minimum skew and jitter will then be

$$t_{margin} = jitter_{PLL} + jitter_{buffer} + skew_{buffer} + skew_{local} \tag{9.6}$$

As SoC processors increase in complexity and size, clock wire length increases proportionally. As the equations above show, this will also increase the absolute value of the timing margin needed for skew and jitter. SoC processor frequencies, however, also increase, which inevitably results in a large increase in timing margin needed for skew and jitter. This undoubtedly necessitates for active skew reduction techniques.

9.4 Active Skew Management Strategies

Active skew management has proven to be an effective skew reduction strategy as microprocessor speeds soared into the hundreds of megahertz range. Early forms of skew reduction mechanisms consisted of designing well balanced clock tree structures such as binary trees, grids, H-trees, and spines [1]. Although regular tree structures significantly contributed to clock skew reduction, load mismatches and intrachip process, temperature, and voltage (PVT) variation both contributed to an increase in clock skew. The percentage of clock period lost to PVT variation increases with operating frequency.

9.4.1 Effect of Clock Skew & Jitter on Power Consumption

Jitter and clock skew is important not only to obtain the maximum possible clock speeds, but also to minimize power consumption. In many DSP applications, there is little advantage to increasing the clock speed beyond a certain sampling rate. Doing so would only incur extra power consumption. This is because the dynamic power consumption of a CMOS digital block is given as

$$P_{dyn} = a \cdot C_L V_{dd}^2 f \tag{9.7}$$

where a is the probability of 0→1 logic transition (switching activity), C_L is

the total load capacitance, V_{dd} is the supply voltage and f is the operating frequency. Since the dominant component of power consumption of a CMOS digital block is dynamic power, the total power consumption is linearly related to the operating frequency.

One effective method of reducing power consumption is to scale down the supply voltage. As equation (9.7) indicates, this leads to quadratic reduction in power consumption. Doing so, however, would result in increase circuit delays. Clock skew and jitter subtract directly from the total amount of delay that the processor can have, thus limiting the potential power savings. This is especially true for dynamic voltage scaling systems [16,33-34], where the power supply can be variable. The delay in a CMOS gate can be estimated to be

$$t_d = \frac{C_L V_{DD}}{I_{avg}} \tag{9.8}$$

where I_{avg} is the average dynamic current in the gate. Using the alpha power law for current in a CMOS device [17], the delay can be shown to be

$$t_d = \frac{kV_{DD}}{(V_{DD} - V_T)^{\alpha}} \tag{9.9}$$

where V_T is the device threshold, V_{DD} is the power supply, k is a proportionality constant, and α is a device related constant. For long channel devices ($L > 1\mu m$), α is approximately 2. For short channel devices, α is between 1.0 and 1.5. Assuming that the frequency of the CMOS circuit is tuned to its maximum speed, using (9.7) and (9.9), it can be shown that the power consumption is equal to

$$P_{dyn} = k_2 V_{DD}(V_{DD} - V_T)^{\alpha} \tag{9.10}$$

To maintain the same circuit speed, the supply voltage can be incrementally increased to compensate for clock jitter and skew. Increasing the supply voltage, however, incurs a power penalty. To estimate this power penalty, the relationship between Δt_d and ΔV_{DD} is first established. Using Taylor series approximation, equation (9.9) can be rewritten as

$$t_d = \left.\frac{kV_{DD}}{(V_{DD} - V_T)^{\alpha}}\right|_{V_{DD}=V_{DD0}} \approx \frac{kV_{DD0}}{(V_{DD0} - V_T)^{\alpha}} + kV_{DD}\frac{V_{DD}\cdot(1-\alpha) - V_T}{(V_{DD0} - V_T)^{\alpha+1}} \tag{9.11}$$

Only a linear approximation of delay was taken in (9.11). This is valid since the clock jitter and skew is expected to be a small fraction of the total clock period. Using (9.11) it can be shown that the incremental voltage to compensate for clock jitter and skew of Δt_d is given as

$$\Delta V_{DD} = \frac{\Delta t_d \cdot (V_{DD0} - V_T)^{\alpha}}{V_{DD0} \cdot (1-\alpha) - V_T} \tag{9.12}$$

Next, the linear Taylor series approximation of (9.10) is derived and the incremental power increase for a given voltage increment is given as

$$\Delta P_{dyn} = k_2 \cdot (\Delta V_{DD}) \cdot (V_{DD0} - V_T)^{\alpha-1} \cdot [V_{DD0}(1+\alpha) - V_T] \tag{9.13}$$

Substituting (9.12) into (9.13) gives

$$\Delta P_{dyn} = k_2 \cdot \left(\frac{\Delta t_d}{k[V_{DD0} \cdot (1-\alpha) - V_T]} \right) \cdot (V_{DD0} - V_T)^{2\alpha} \cdot [V_{DD0}(1+\alpha) - V_T] \tag{9.14}$$

To understand the dynamics of equation (9.14) it was plotted as shown in Fig. 9-7. Four sets of values were used and the Δt_d values from –5ps to +5ps were used for jitter and skew values. Since linear Taylor series was used, it is expected that this graph would be linear. Curves Dp1 and Dp3 have V_{DD}/V_T ratios of 3, and Dp2 and Dp4 have V_{DD}/V_T ratios of 2. Curves Dp1 and Dp2 have α coefficient of 1.3, and curves Dp3 and Dp4 have α coefficient of 1.8. As the figure shows, smaller V_{DD}/V_T ratios result in less power consumption penalty. This is because according to equation (9.9), the delay is more sensitive to the power supply in this regime. This means that it takes less supply voltage change to cause large reduction in delay. The curve also shows that reduced value of α causes an increased sensitivity of the power consumption increment to delay variation. This, again, has to do with the sensitivity of delay to α as shown in equation (9.9)

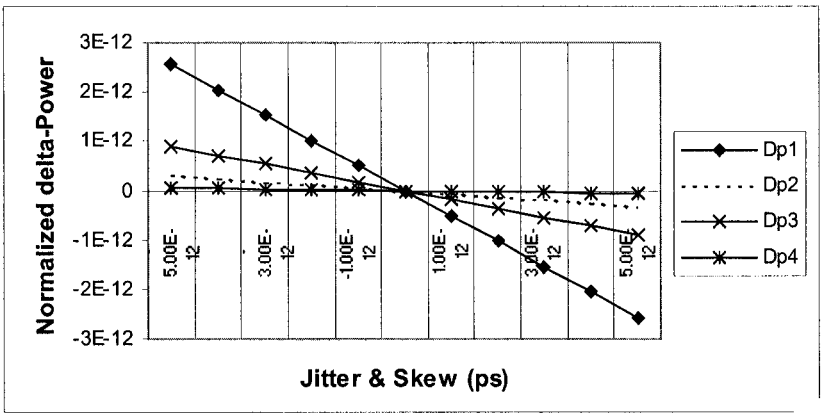


Figure 9-7. Dynamics of increase in power consumption due to clock skew and jitter

9.4.2 Ad-Hoc Deskew Strategies

To solve the problem of clock skew without having to alter the supply voltage, a variety of active skew management techniques have been reported in the literature. One technique commonly used in processors running at several hundred megahertz clock rates is based on a PLL feedback technique, as shown in Fig. 9-8 [6]. In this technique, a point in the middle of the clock tree is chosen and used as a feedback signal back into the PLL. This has the effect of reducing the clock skew by half. The disadvantage of this technique is that it cannot be extended for very large die areas and skew can only reduced by a factor of 2. One variation on this technique uses distributed oscillators [7]. The advantage of this technique is that a DC control voltage is used for global distribution instead of a high-frequency clock signal. This technique, however, can potentially suffer from large jitter due to the control voltage high sensitivity to supply noise and coupling noise from nearby signals.

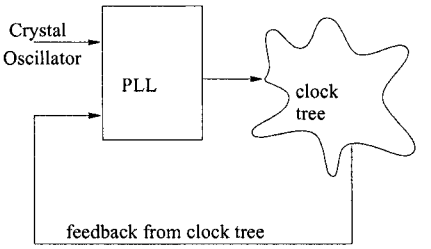


Figure 9-8. PLL feedback resynchronization technique

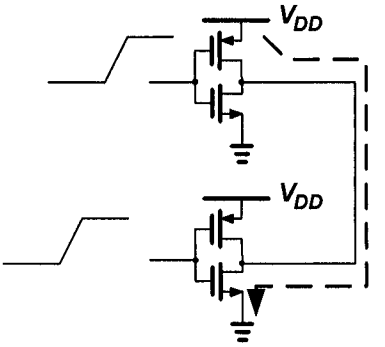


Figure 9-9. Short-circuit current path resulting from skew averaging technique

Another technique relies on distributing the clock in a grid configuration and shorting all ends of the global distribution line [8]. This has the effect of

averaging out the skew between different regions across the chip. The disadvantage of such a technique is that it results in larger power consumption due to increased short-circuit current. Short circuit current results from two buffers with shorted outputs and skewed input clock signals, as illustrated in Fig. 9-9. The short-circuit current can increase clock power dissipation by 20% - 30%.

9.4.3 DLL-based Deskew Strategies

Another class of more powerful techniques rely on delay locked loops, or DLLs [27-32]. One such topology [9] is shown in Fig. 9-10. This technique is also considered to be a feedback technique. In this technique, the digital design is broken up into deskew clusters. Each cluster receives its clock signal through up to four deskew registers (DSK). The clock signal is generated by a PLL, which produces an integer multiple of a high precision crystal oscillator. A deskew register consists of a digitally controlled delay cell and a phase detector. The two inputs to the phase detector are the clock signal and the crystal oscillator signal. Skew is minimized by delaying the crystal oscillator signal and compensate for it by varying the delay of the clock signal at the deskew buffer. In effect, the skew contribution now comes from the skew in the clock reference signal and the clock uncertainty in the deskew buffer's phase detector, as opposed to skew due to clock network due to PVT and load mismatches. In general, this has resulted in reduced skew.

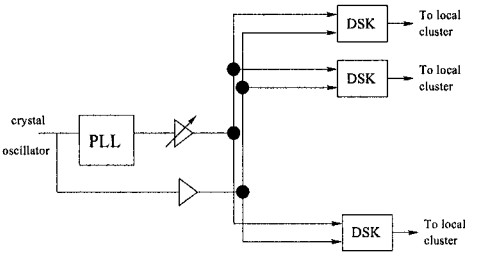


Figure 9-10. Active skew reduction using DLL approach

The skew reduction in the previous technique, however, suffers from large variation in performance due to intrachip PVT. To address this issue, another technique has been recently suggested in [10]. As with the register deskew technique, the digital design is broken up into clusters. Each cluster receives the global clock signal through a digitally programmable delay cell. In order to compensate for skew due for PVT and capacitive load mismatches, on-chip fuses are blown during factory test. The advantage of

this technique compared to the DLL approach listed above is that each cluster's deskew register can have its own setting depending on its local process variation. Voltage and temperature are still uncompensated.

9.5 Multi-Phase Clock Generator

In the previous section, various techniques in eliminating clock skew have been investigated. In some applications, however, intentional clock skew is required. For example, static random access memories (SRAMs) normally require multiple phases of the clock to perform the various tasks required for its operation [11,21-24]. Moreover, tight control of the phases of the clock is required for high-speed operation. Other applications that require multi-phase clock generation include wave-pipelined dynamic logic [18,35] and parallel sigma-delta A/D converters [19].

9.5.1 Passive Method

The simplest technique to generate multiple phases of the clock is known as the *passive technique*. In this technique, a ring oscillator with multiple stages is designed and the various phases of the ring oscillator are used as shown in Figure 9-11. No additional circuitry is required to generate different phases of the clock. In order to prevent variation with PVT, the ring oscillator is usually operated in a phase-locked loop. This technique may suffer from mismatches in the generated phases due to device mismatches in the ring oscillator's delay cells.

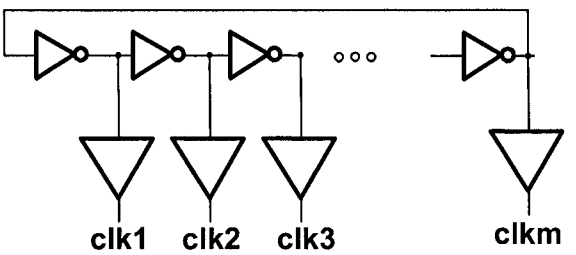


Figure 9-11. Passive multi-phase clock generator

9.5.2 Master-Slave DLL

Another technique of generating multiple phases of the clock is to use the *master-slave DLL* approach [19]. Fig. 9-12 shows the overall schematic diagram of this approach. A single master DLL is used to generate a pulse at the negative edge of the input clock. The pulse width of this negative pulse

is used to generate a common reference timing pulse to all slave DLLs. The delay of each slave DLL is tuned individually through programming bits on a current DAC. This provides digital control on the input current to the slave DLL, and hence its delay. The multi-phase raw clock-generator (MRPCG) is a passive clock generator as shown in Fig. 9-11. The slave DLLs improve the accuracy and skew of the passive clock generator.

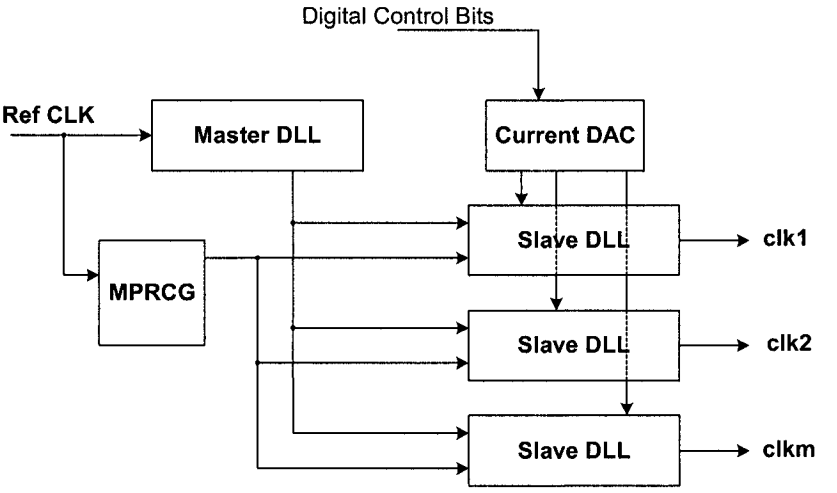


Figure 9-12. Schematic diagram of master-slave DLL approach

Fig. 9-13 shows a schematic diagram of a master DLL. The charge pump either injects current or draws current from the loop filter capacitor C_p. At steady-state, the amount of charge drawn from C_p during one period is equal to the amount of charge injected into C_p. The pole introduced by R_L and C_L limit the magnitude of the ripple at the charge pump output net.

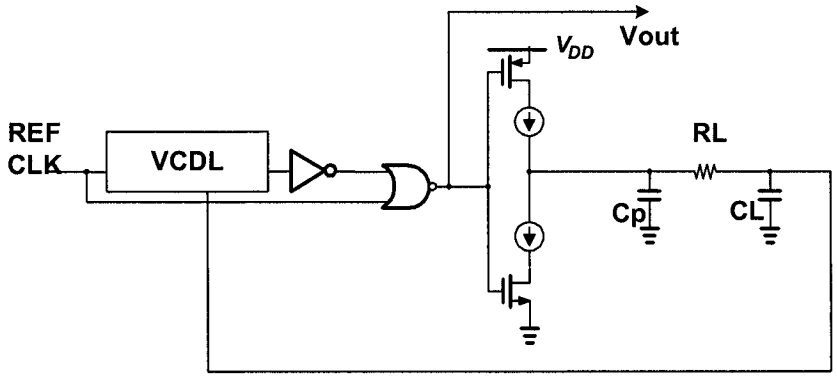


Figure 9-13. Master DLL in master-slave DLL approach

The slave DLL is shown in Figure 9-14. The master DLL controls the DOWN pulse width, and hence the amount of charge periodically drawn from C_p. The UP pulse is controlled by the slave DLL's delay line as well as the multi-phase raw clock generator (MPRCG). As in the master DLL, the net charge displacement across C_p during one period should be zero. The means that steady-state,

$$\frac{I_{up}}{C_p} t_d = M \frac{I_{down}}{C_p} \cdot \frac{I_{up}}{I_{up} + I_{down}} \cdot T \tag{9.15}$$

where T is one reference period, I_{up} is the UP current and I_{down} is the DOWN current, M is the number of phases generated by the MPRCG, and t_d is the delay generated by the slave DLL. Solving (9.15) for t_d gives

$$t_d = M \frac{I_{down}}{I_{up}} T_0 \tag{9.16}$$

where T_0 is given as

$$T_0 = \frac{I_{up}}{I_{up} + I_{down}} T \tag{9.17}$$

If the current DAC is added to the down current, the delay increment due to change in DAC current is given as

$$\Delta t_d = M \frac{\Delta I_{DAC}}{I_{up}} T_0 \tag{9.18}$$

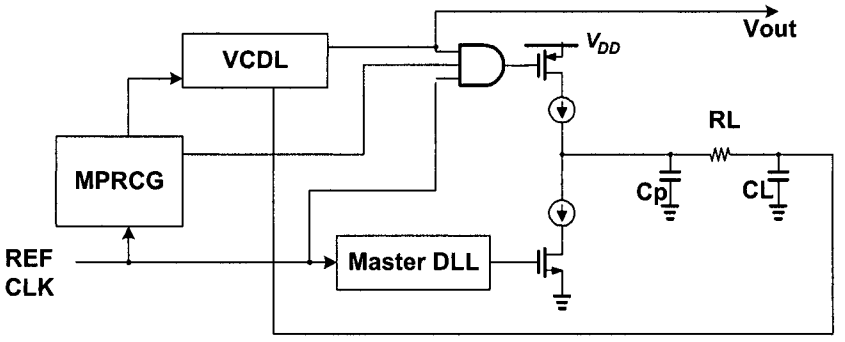


Figure 9-14. Slave DLL in master-slave DLL approach

This means that the resolution of the delay is proportional to the minimum achievable current increment, ΔI_{DAC}, and reference period, T_0. Using this technique, a timing resolution of 2ps/uA and a power supply sensitivity of 0.015ps/mV have been demonstrated in a 0.3um CMOS technology [19]. Using a regular passive method, the power supply sensitivity was 0.375ps/mV.

9.5.3 Pulsewidth Control Loops

The above technique showed how to generate multiple phases of an input clock without any control on the duty cycle of the generated clock phases. In applications where the target circuit operates only on edges, this technique should be sufficient. However, in many other cases, control of duty cycle is important. For example, modern high-speed pipelined digital systems or digital circuits employing dynamic logic operate on both the rising and falling edges of the clock. It is common for these applications to have an imbalance in the timing requirement between circuitry operating on the positive and negative edges of the clock. For such applications, control of the duty cycle is important.

Fig. 9-15 shows a technique of accurately generating digitally controlled duty cycles of an input clock signal [20]. Although not required for correct operation, the circuit works best when the input clock signal has 50% duty cycle. The operation of the circuit is very similar to the slave DLL shown previously. Instead of a DLL, two delay lines are used and their outputs are NAND'ed together. The first is a fixed delay line and the second is a voltage-controlled delay line (VCDL), as shown in Fig. 9-15. The delay difference in the two delay lines generates a pulse width proportional to that delay difference. The charge pump is the same as the phase detector in the master or slave DLL previously shown.

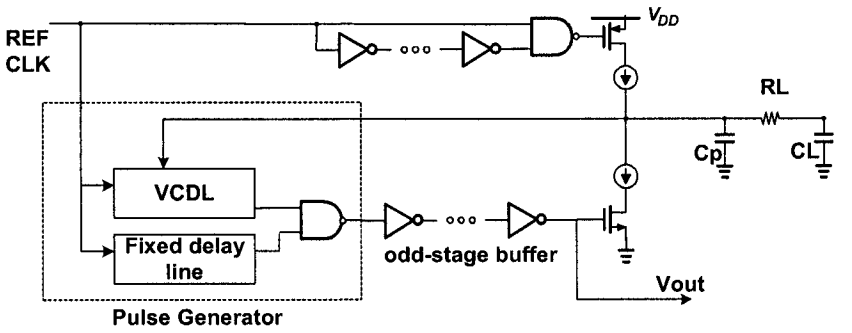


Figure 9-15. Schematic diagram of fixed pulse-width control loop

9.6 Low-Power Clock Distribution Strategies

Power consumption is also a major concern in clock distribution networks. The most common method of reducing power in a clock distribution network is by using clock gating [11]. Clock gating typically saves around 30% of power consumption. As clock speeds soar to the gigahertz range, clock gating alone is insufficient to maintain the power budget of the clock distribution network under control. More elaborate techniques need to be explored.

9.6.1 Active Clock Gating

Ever since the early days of low-power CMOS design, one of the most well-established power reduction techniques has been quoted as being clock gating. As stated earlier, the dynamic power consumption of a CMOS digital design is given as

$$P_{dyn} = a \cdot C_L V_{dd}^2 f \tag{9.19}$$

where a is the probability of 0→1 logic transition (switching activity), C_L is the total load capacitance, V_{dd} is the supply voltage and f is the operating frequency. When a certain unit in a SoC is not being used, unnecessary toggling can be inhibited by simply gating off the clock signal [11]. Assuming the unit is a synchronous unit, no dynamic power is consumed by the block. Optimal control of the clock gating signals by software or the central processing unit (CPU) has been a subject of intense research in the recent past. Although effective, the average clock routing power in a SoC processor can be reduced by only 30% using clock gating.

9.6.2 Low-power Gigahertz Clock Strategies

As clock frequencies increase to beyond the gigahertz range, new techniques for saving power consumption present themselves [25]. One such technique [12] attempts to recycle the reflected clock signal to save power consumption. Up to 60% power savings have been reported using this technique. Another implementation [13] attempts to create a standing wave across the clock distribution net and resonate at the desired frequency. Ideally, there would be no power loss due to the impedance of the clock network impedance. Both techniques, however, rely on the ability to produce very low impedance clock drivers as well as careful control of the impedance of clock distribution networks. Also, in order to create a standing wave a cross-coupled NFET pair is needed in order to compensate for

resistive losses in the clock network. More specifically, in order to create a standing wave, these two conditions must be satisfied:

$$g_{ccp} < \frac{1}{n} \cdot \frac{RC}{L} \tag{9.20}$$

and

$$l = \frac{1}{2f_{clk}\sqrt{LC}} \tag{9.21}$$

The inductance and capacitance are controlled by having on-chip spiral inductors and varactors at the end of a clock distribution cluster. Accurate control of the varactor over PVT is still subject to further investigation. Moreover, this strategy saves power on the global clock network. Studies have shown that as much as 90% of the clock power of modern processors is dissipated in the local clock networks [9].

9.7 Multi-clock Domain Interfacing

As SoC processors become more heterogeneous, so does their clock frequencies and requirements. In some instances, data must be passed from one block to another in a SoC processor, each with its own operating clock frequency. For this reason, special synchronization techniques are required to interface the two blocks [26].

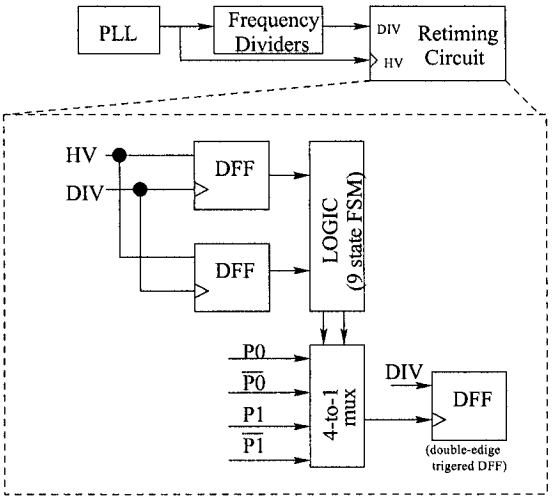


Figure 9-16. Block diagram of the proposed retiming circuit

In this section, an all-digital synchronization is introduced which describes how to synchronize between two asynchronous digital cores in an

SoC processor. The basic idea behind this technique is to produce four low-jitter phases of the input clock signal. One of the phases is then chosen to retime the input signal. This retiming operation, which is done by the final stage flip-flop, is what synchronizes that input data with the local clock signal. Four phases of the input clock is necessary in order to choose an input clock phase that has its edges sufficiently far away from the output clock edges. This is important in order to avoid timing hazards at the final stage flip-flop. Fig. 9-16 shows a block diagram of the proposed retiming operation. Table 9-1 shows the decoding logic required to choose which phase to retime the divided down clock signal with.

Table 9-1. Decoding Logic used in ADCR

DFF Values	Phase choice
11	$\overline{P0}$
00	P0
10	P1
01	$\overline{P1}$

Fig. 9-17 shows a timing diagram illustrating the operation of the All-Digital Clock Retiming (ADCR) technique. The phase for retiming is chosen in such a way that it is the soonest phase *after* the input (output of divider) arrives. The input, in this case, is the divided down jitter clock signal (can also be data from another unit in the SoC). The block labeled "LOGIC" is actually a finite state machine. This state machine ensures that if a clock phase is initially chosen and later found to be close to the arrival edge of the data signal, then the next clock phase is chosen. The state machine ensures that the change in phase selection only occurs once. This is necessary in order to avoid switching of the selection of the retiming phase, which would induce jitter. The final stage flip-flop performs the actual retiming of the clock edge. This is a double edge triggered flip-flop (DETFF). A DETFF is necessary in order to prevent distortion of duty cycle of the retimed signal.

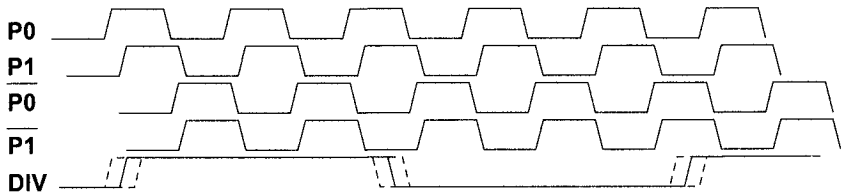


Figure 9-17. Timing diagram of retiming clock edges and frequency divided down signal

The finite state machine of the ADCR can be simplified by noticing that since the DETFF operates on both edges of the input clock, it only requires 2 phases (the other 180 degrees phases are implied by the operation of the flip-flop). One phase would be the original clock signal and the other phase is a delayed version of the original clock signal. The only restriction on the delay is that it should be greater than the expected jitter, but less than half the maximum PLL output period.

Reducing the number of phases required to two phase simplifies the finite state machine. The FSM can be reduced from 9 states down to 5 states, as shown in Fig. 9-18. This also corresponds to hardware reduction (3 flip-flops instead of 4, and reduced combinatorial logic). Fig. 9-19 shows the impact of the retiming circuit. As the figure shows, it completely eliminates the jitter caused by the noisy digital dividers, or uncertainty in input signal arrival time.

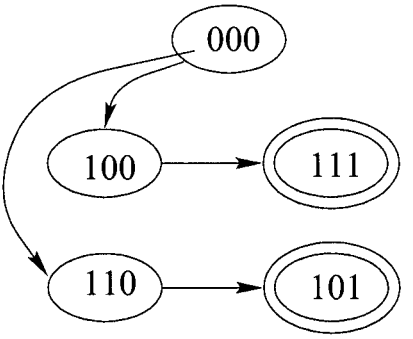


Figure 9-18: Finite state machine of reduced state ADCR circuit

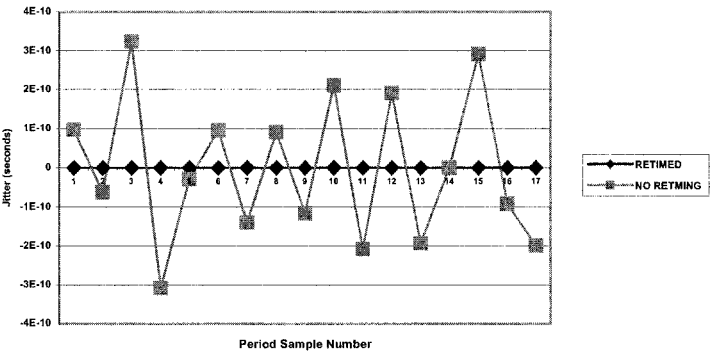


Figure 9-19: Comparison of jitter clock and retimed clock signals

9.8 Summary

Techniques for multi-gigahertz clock routing for SoC processors was reviewed. The issue was presented from a chip-level point of view. Basic device theory was reviewed to present a model for clock distribution lines. Based on this, theoretical limits in CMOS nanometer technologies on achieving reliable jitter and skew budgets was derived. An overview was then presented on some recent techniques of reducing clock skew. Trends still show that clock distribution will only continue to consume a larger percentage of the total clock period if more aggressive deskewing strategies are not developed. All deskew and low-power clock routing techniques discussed in sections 9.4 and 9.6 are summarized in Table 9-2.

Table 9-2. Comparison of Clock Deskewing Approaches

Approach	**Clock Frequency (GHz)**	**% of period lost to jitter & skew**
PLL feedback [6]	0.3	4.62
DLL [9]	0.8	4.16
Fuse-based [10]	1.5	3.60
Shorting grid [8]	5.0	5.00
Wave recycling [12]	5.0	10.00
Standing wave [13]	10.0	3.20

Multi-phase clock generation and interfacing between different clock regimes in SoC processors was also covered in this chapter. Two different multi-phase clock generations techniques were introduced: Master-Slave DLL approach and the pulse width control technique. Both improve over the passive technique of multi-phase generation by having a multi-delay element ring oscillator. An all-digital clock resynchronization algorithm was also introduced in the final section of this chapter.

REFERENCES

[1] E. Friedman, "Clock Distribution Networks in Synchronous Digital Integrated Circuits," Invited paper, *Proc. Of the IEEE*, No. 5, May 2001, pp. 3-14.

[2] R. Y. Chen, et. al., "Clock Power Issues in System-on-a-Chip Designs," *IEEE Workshop on VLSI*, pp. 48-53.

[3] P.K. Chatterjee, et al., "The Impact of Scaling Laws on the Choice of N-Channel or P-Channel for MOS VLSI," *IEEE Electron Device* Letters, vol. EDL-1, pp. 20-223, October 1980.

[4] X. Huang, et al., "Loop-Based Interconnect Modeling and Optimization Approach for Multigigahertz Clock Network Design," *J. of Solid-State Circuits*, vol. 38, no. 3, pp. 457-463, March 2003.

[5] G. Gonzalez, *Microwave Transistor Amplifiers: Analysis and Design*, Prentice-Hall: New Jersey, 1997.

[6] E. Friedman, "Clock Distribution Networks in Synchronous Digital Integrated Circuits," Invited paper, *Proc. Of the IEEE*, No. 5, May 2001, pp. 3-14.

[7] I. Young, et al., "A 0.35um CMOS 3-880MHz PLL N/2 clock multiplier and distribution network with low jitter for microprocessors," *ISSCC* Feb. 1997, pp. 330-331.

[8] H. Mizuno, K. Ishibashi, "A Noise-Immune GHz-Clock Distribution Scheme Using Synchronous Distributed Oscillators," *ISSCC*, Feb. 1998, pp. 404-405.

[9] S. Tam, et al., "Clock Generation and Distribution for the First IA-64 Microprocessor," *IEEE JSSC*, vol. 35, Nov. 2000, pp. 1545-1552.

[10] S. Tam, et al., "Clock Generation and Distribution for the 130-nm Itanium 2 Processor with 6-MB On-Die L3 Cache," *IEEE J. of Solid-State Circuits*, vol. 39, no. 4, April 2004, pp. 636-642.

[11] A. Bellaouar, M. I. Elmasry, *Low-Power Digital VLSI Design: Circuits and Systems*, Kluwer Academic Publishers: Boston, 1996.

[12] M. Mizuno, "On-Chip Multi-GHz Clocking with Transmission Lines," *ISSCC*, 2000, pp. 336-337.

[13] F. O'Mahony, et al., "10GHz Clock Distribution Using Coupled Standing-Wave Oscillators," *ISSCC*, Feb. 2003, pp. 428-504.

[14] A. Mule, et al., "Electrical and Optical Clock Distributino Networks for Gigascale Microprocessors," *IEEE Transactions on* VLSI, vol. 10, no. 5, Oct. 2002, pp. 582-594.

[15] W. Liu, *MOSFET Models for SPICE Simulation including BSIM3v3 and BSIM4*, Wiley: New York, 2001.

[16] M. Elgebaly, A. M. Fahim, I. Kang, and M. Sachdev, "Efficient Dynamic Voltage Scaling Architecture," *IEEE International SOC Conference*, pp. 155-158, September 17-20, 2003.

[17] T. Sakurai and R. Newton, "Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas," *IEEE J. of Solid-State Circuits*, April 1990.

[18] W. Lien and W. Burleson, "Wave-domino logic: Theory and Applications," *IEEE Transactions on Circuits and Systems II*, vol. 42, no. 2, pp. 78-91, 1995.

[19] A. Suzuki, et. al., "A Digitally Skew Correctable Multi-Phase Clock Generator Using a Master-Slave DLL," *Int'l Symposium on Circuits and Systems*, vol. 1, pp. 105-108.

[20] P. Yang and J. Wang, "Low-Voltage Pulsewidth Control Loops for SoC Applications," *IEEE J. of Solid-State Circuits*, vol. 37, no. 10, pp. 1348-1351, October 2002.

[21] T. Matano, et. al., "A 1-Gb/s/pin 512-Mb DDRII SDRAM using a digital DLL and slew-rate-controlled output buffer," *Symposium on VLSI Circuits*, pp. 112-113, 2002.

[22] J. Lee, et. al., "Digitally-Controlled DLL and I/O Circuits for 500Mb/s/pin x16 DDR SDRAM," *Int'l Solid-State Circuits Conference*, pp. 68-69, 2001.

[23] T. Hamamoto, et. al., "A Skew Jitter Suppressed DLL Architecture for high frequency DDR SDRAMs," *Symposium on VLSI Circuits,* pp. 76-77, 2000.

[24] J. Kim, et. al., "A Low-Jitter Mixed-Mode DLL for High-Speed DRAM Applications," *IEEE J. of Solid-State Circuits*, vol. 35, no. 10, pp. 1430-1436, October 2000.

[25] S. Chan, "A 4.6GHz Resonant Global Clock Distribution Network," *Int'l Solid-State Circuits Conference*, pp. 342-343, 2004.

[26] I. Brynjolfson and Z. Zilic, "MCSoC: A Platform for Clock Managed Systems on a Chip," *IEEE Custom Integrated Circuits*, pp. 117-121, 2001.

[27] J. Park, Y. Koo, and W. Kim, "A Semi-Digital Delay Locked Loop for Clock Skew Minimization," *12th International Conference on VLSI Design*, pp. 584-588, 1999.

[28] H. Lee, H. Nguyen, D. Potter, "Design of Self-Synchronized Clock Distribution Netowrks in an SoC ASIC Using DLL with Remote Clock Feedback,"*Int'l ASIC/SOC Conference*, pp. 248-252, 2000.

[29] A. Waizman, "A Delay Line Loop for Frequency Synthesis of De-Skewed Clock," *Int'l Solid-State Circuits Conference,* pp. 298-299, 1994.

[30] G. Dehng, et. al., "Clock-Deskew Buffer Using a SAR-Controlled Delay-Locked Loop," *IEEE J. of Solid-State Circuits*, vol. 35, no. 8, pp. 1128-1136, August 2000.

[31] S. Liu, J. Lee, and H. Tsao, "Low-Power Clock-Deskew Buffer for High-Speed Digital Circuits," *IEEE J. of Solid-State Circuits*, vol. 34, no. 4, pp. 554-558, April 1999.

[32] S. Tanoi, et. al., "A 250-62 MHz Deskew and Jitter-Suppressed Clock Buffer Using Two-Loop Architecture," *IEEE J. of Solid-State Circuits*, vol. 31, no. 4, pp. 487-493, April 1996.

[33] S. Akui, et. al., "Dynamic Voltage and Frequency Management for a Low-Power Embedded Microprocessor," *Int'l Solid-State Circuits Conference*, pp. 64-65, 2004.

[34] C. Lichtenau, et. al., "PowerTune: Advanced Frequency and Power Scaling on 64b PowerPC Microprocessor," *Int'l Solid-State Circuits Conference*, pp. 356-357, 2004

[35] K. Nose and M. Mizuno, "Parallel Clocking: A Multi-Phase Clock-Network for 10GHz SoC," *Int'l Solid-State Circuits Conference*, pp. 344-345, 2004.

Index

Analog IP 215
A/D converter 164, 166
Analog oscilloscope 204
Autocorrelation function 168
Automated test equipment 204

Bandgap voltage 50-50
Barkhausen's criteria 68
Bessel function 66
Boltzmann constant 81

Capacitor array 191
Carry look-ahead adders 147
Charge pump
  Current steering 39
  Low-voltage 40
Chi squared statistic 208
Clock skew 219, 220, 224, 225
CMOS
  Capacitor 29-30
  Device noise 33
  Threshold voltage 28
  Current, saturation 30
  Current, linear 30
  Transconductance 31
  Small-signal model 32
Common mode feedback 100
Control voltage 18-21, 103, 104, 125, 133
Counter 12-13
Counter, MN 185
Crosstalk 202
Cumulative density function 214
Current controlled oscillator 7
Current references
  PTAT current reference 54-56, 93
  CTAT current reference 56
  Constant-gm 150
  Bandgap 56
  Low-voltage 57
Current mirrors 33-37
Cyclostationary phase noise 68, 72-75
Cycle slipping 19

Damping factor 14, 21, 124, 125, 144, 145
Dead zone, phase detector 9, 10, 87
Delay locked loop (DLL) 129-134, 229-230

Deskew buffer 229
Deterministic jitter 38, 40, 63-64, 205, 207, 208
Digital oscilloscope 202
Differential PLL 100-107
Digital loop filter 142-145
Digital-to-analog converter 147-148, 149, 169, 185
Digital signal processor 164, 165, 194
Dirac function 171
Direct digital synthesizer 185
Divider,
  Frequency 12-13
  Noise 172-177
  Pre-divider 178
  Post-divider 178
  Programmable 9, 13
Double edge-triggered flip-flop 236
DRAM 6
Dynamic voltage scaling 225

Emitter coupled logic 202

Finite Impulse Response 164
Flicker noise 32, 33, 61, 71, 75, 84, 85
Folded cascode 111
Fourier Series 73
Fuses 229
Fractional-N PLLs 165, 178-179
Frequency accuracy 3, 5, 198-199
Frequency locked loop 156-159
Frequency test 198-199
FM modulation 66

Impulse sensitivity function 73
  Parasitic trace inductance 221-223
  I/O activity 200

Jitter,
  Absolute 62, 152
  Adjacent cycle jitter 64
  Cycle-to-cycle 64, 186, 187
  Deterministic 67, 205, 206, 208
  Edge-to-edge 64
  Period 61
  Random 63, 207, 208

Kalman Filter 127

Leeson formula 70
Leeson-Cutler formula 71
Line proximity effect 221
Load board 202, 215
Lock detector 200
Lock-in process 18-25
Lock time 4, 5, 20, 88, 123, 124, 156, 157, 178, 193, 198, 199
Locking behavior 18-25
Loop filter 3, 4, 6, 7, 10-11, 15, 142, 144
Low voltage
  Charge pump 40
  VCO 40-44

Maneatis loads 50, 148
Multiply-and-accumulate 164
Multi-phase raw clock generator 232

Noise,
  Flicker *see flicker noise*
  Ground bounce 75, 92
  Power supply 89-91, 107
  Substrate 92
  Thermal 33
Nyquist rate 169

Oversampling 185

Package 200
Pad rings 201

Phase detector
  Dead zone *see dead zone*
  XOR 7
  Three-state 8
  Type-4 10
PFD nonlinearity 86-87
Phase noise
  Power supply 89-91, 107
  VCO 84
Power consumption 71
Power supply rejection 107-111
Process insensitive design 124-126

Q, resonant VCO 69

Range, phase detector 88
Resistorless loop filter 128
Ring oscillator 68, 73, 92, 75-79

Sampling rate 164
Self-biased PLL 125
Self resonance 224
Sigma-delta 169-171
Signal to noise ratio (SNR) 164
Skin effect 221-222
SMA connectors 203
Spurious free dynamic range 164, 193

TailFit 207, 208
Tektronix 208
Time interval error (TIE) 62
Two mode bandwidth 123, 126

Vernier delay line (VDL) 214
Voltage Controlled Oscillator
  LC tank 79-82
  Ring oscillator 68, 73, 92, 75-79
Voltage regulator 112
Voltage-to-current converter 7, 12, 44
Voltage control delay line 129, 130

Wave pipelining 230

Z-domain 144